UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

FACULTAT D'INFORMÀTICA DE BARCELONA (FIB)

Bachelor Degree in Informatics Engineering

Specialization in Computing

Celebrity Games and Critical Distance

Supervisor:
Àlvarez Faura, Carme
DEPARTMENT OF COMPUTER SCIENCE

Author:
Bofarull Cabello, Antoni

Academic Year 2018/2019
Date 04/07/2019

Abstract

In this project we study the dynamic behavior of Sum Celebrity and Max Celebrity Games defined in Àlvarez et al. 2016 and Àlvarez & Messegué 2016, respectively. These works analyze the structural properties of the Nash equilibrium graphs and the relationship between the social cost and the optimal social cost (Price of the Anarchy and Price of Stability) according to the parameters that define these games: alpha or cost of each link, beta or critical distance and weights of each of the players.

In this project we analyze different dynamics of the Sum and Max Celebrity Games models and their convergence to equilibrium configurations as a function of the critical distance. A dynamics consists of a sequence of movements where both the policy to select the player and the kind of selfish strategy that the corresponding player applies, can affect convergence. The starting point is that in both models the Best Response strategy of a player is computable in polynomial time for $\beta = 1$, but NP-hard for $\beta > 1$.

Due to this difference in behavior, both cases have been analyzed separately. For $\beta = 1$ we have obtained that for both models the problem of computing a Nash equilibrium is polynomial time. Furthermore, in the case of Max we have proven the existence of cycles in a Best Response dynamics, in contrast to the Sum where they are not possible. Since the problem of computing a Best Response for both models when $\beta > 1$ is NP-hard, we have proposed a greedy model in which the possible strategies of a player are restricted. Although the existence of cycles is proven, we have not found experimentally a cyclic instance for the Sum and Max Greedy Celebrity Games.

Resum

En aquest projecte estudiem el comportament dinàmic del Sum Celebrity i Max Celebrity Games definits per Àlvarez et al. 2016 i Àlvarez & Messegué 2016, respectivament. Aquests treballs analitzen les propietats estructurals dels equilibris de Nash junt amb la relació entre el cost social i el cost social òptim (Preu de l'anarquia i Preu de l'estabilitat) d'acord amb els paràmetres que defineixen aquests jocs: alfa o cost de l'enllaç, beta o distància crítica i els pesos de cadascun dels jugadors.

En aquest projecte analitzem les diferents dinàmiques dels models del Sum i Max Celebrity Games i la seva convergència a configuracions d'equilibri en funció de la distància crítica. Una dinàmica consisteix en una seqüència de moviments on la política de selecció del jugador i el tipus d'estratègia egoista que el jugador corresponent aplica, pot afectar la convergència. El punt de partida és que en ambdós models calcular la Best Response per a un jugador és computable en temps polinòmic per $\beta = 1$, però NP-hard per $\beta > 1$.

A causa d'aquesta diferència en el comportament, els dos casos han estat analitzats per separat. Per $\beta = 1$ hem obtingut que per a tots dos models el problema de calcular un equilibri de Nash és temps polinòmic. A més, en el cas del Max hem demostrat l'existència de cicles en una dinàmica en la qual s'aplica el Best Response, en contraposició amb el Sum on no són possibles. Atès que el problema de calcular la Best Response per a tots dos models quan $\beta > 1$ és NP-hard, hem proposat un model greedy en què les possibles estratègies d'un jugador estan restringides. Tot i que està demostrat l'existència de cicles, no hem trobat experimentalment una instància cíclica per als jocs Sum i Max Greedy Celebrity Games.

Resumen

En este proyecto estudiamos el comportamiento dinámico del Sum Celebrity y Max Celebrity Games definidos por Álvarez et al. 2016 y Álvarez & Messegué 2016, respectivamente. Estos trabajos analizan las propiedades estructurales de los equilibrios de Nash junto con la relación entre el coste social y el coste social óptimo (Precio de la anarquía y Precio de la estabilidad) de acuerdo con los parámetros que definen estos juegos: alfa o coste del enlace, beta o distancia crítica y los pesos de cada uno de los jugadores.

En este proyecto analizamos las diferentes dinámicas de los modelos del Sum y Max Celebrity Games y su convergencia en configuraciones de equilibrio en función de la distancia crítica. Una dinámica consiste en una secuencia de movimientos donde la política de selección del jugador y el tipo de estrategia egoísta que el jugador correspondiente aplica, puede afectar la convergencia. El punto de partida es que en ambos modelos calcular la Best Response para un jugador es computable en tiempo polinómico para $\beta = 1$, pero NP-hard para $\beta > 1$.

Debido a esta diferencia en el comportamiento, ambos casos han sido analizados por separado. Para $\beta = 1$ hemos obtenido que para ambos modelos el problema de calcular un equilibrio de Nash es tiempo polinómico. Además, en el caso del Max hemos demostrado la existencia de ciclos en una dinámica en la que se aplica el Best Response, en contraposición con el Sum donde no son posibles. Dado que el problema de calcular la Best Response para ambos modelos cuando $\beta > 1$ es NP-hard, hemos propuesto un modelo greedy en el que las posibles estrategias de un jugador están restringidas. Aunque está demostrado la existencia de ciclos, no hemos encontrado experimentalmente una instancia cíclica para los juegos Sum y Max Greedy Celebrity Games.

Contents

1 Introduction

In this chapter we explain the motivation in conducting research, the context of the project, the objectives and how the manuscript has been structured.

1.1 Motivation and Context

Game Theory is a branch of mathematics whose aim is the study of models where a set of agents interact with each other following some objectives or interests. These models usually try to be as simple as possible but whose results can be applied in the real world. Since the appearance of the Internet, there has been a great interest in analyzing communication networks. This interest comes from many diverse scientific communities: computer science, artificial intelligence, networking and economics among others.

Those networks are built by selfish agents in a decentralized way without coordination among the agents, where each of them wants to improve their connectivity using as little investment as possible. This fact was an obstacle to perform the analysis using the classic *Network theory*. A common way to approach the problem is by using Game Theory, as we have already mentioned, deals with problems of similar nature. The concrete area that intersects Game Theory and Computer Science is called *Algorithmic Game Theory*. From an Algorithmic Game Theory point of view, the internet-like networks can be considered as a state of equilibrium of a strategic game played by selfish agents.

Strategic Games model the interaction between a set of players. In these games, each player has a set of possible actions where the preference over one or the other is affected by the actions chosen by the other players. In this interaction between players the network can stabilize. This situation in which no player has incentive to change the strategy is called Nash equilibrium. Once it is defined how the social cost is calculated, in this interaction there will be configurations that are socially optimal, that is, the social cost is minimized. The ratio between the maximum social cost of the Nash equilibria and the optimal social cost is defined as *Price of Anarchy*.

The study of the formation of networks from the point of view of *Strategic Games* is a relatively recent topic. A seminal paper in the study of models trying to explain the behavior of different agents that conform internet, is the one of Fabrikant et al. [1]. The authors introduce by now the classic *Network Creation Games* model (NCG) where selfish agents without central coordination pay for links that they establish unilaterally and takes advantage of the shortest-path to all the nodes. Therefore, the cost of each agent has two components: the cost of the creation of links and the sum of the distances from this agent to the rest. We can consider other cost functions as proposed in Demaine et al. [2], where instead of computing the sum of the distances, the maximum is computed

In this project we focus on *Celebrity Games*. A model recently proposed by Àlvarez et al. [3, 4]. In this interpretation of Network Formation it is considered that not all players are equally important. The relevance of each player is assigned by their weight, the heavier the weight, the more relevant is in the network. Therefore, the rest of players will want to be as close as possible to him. The cost of a player is the sum of the total cost of the links he has bought and the sum of the weights of the players he does not have close enough. To indicate what distance is considered close we use what we call critical distance, represented by β. This parameter indicates that players at a distance less than or equal to the critical distance are considered close enough and therefore they are not computed in the cost of the player in question. We also do the study for the max version introduced by Àlvarez et al. [5].

In this work we study the dynamics of the model. By dynamics we refer to turn the model into a sequential process and focus on the formation of the network through the interaction of selfish agents. In this process time is discretised in turns, where at each turn only one agent can change his strategy. For that reason, we need a *turn policy* that determines how the turns are distributed among the agents. We call *response policy* the way in which a agent determines his new strategy. Since agents act selfishly, the most natural response policy is the Best Response, in which the new strategy is one of those that minimize his cost.

1.2 Project contribution

Understanding the dynamics of decentralized and selfish networks without coordination among agents can help us to understand the behaviour of networks like the Internet. This knowledge can help us to develop better mechanisms that guide agents locally to globally better states.

The analysis of the dynamics in the classic model was made by Lenzner et at. [6, 7]. In this project, the main objective is to perform a similar analysis but for Celebrity Games. As for the classic model, we investigate natural dynamics such as Best Response dynamics.

We know beforehand that the computational complexity of computing a Best Response of a player in the Celebrities depends on the critical distance, represented by β. For that reason we have divided the analysis into $\beta = 1$ and $\beta > 1$.

In the case of $\beta = 1$, we know that the problem of computing a Best Response is polynomial time solvable but we do not know if the dynamics has cycles, that is to say, the players can reach a situation in which they have been previously, and if we can compute a Nash equilibrium in a reasonable time.

Since for $\beta > 1$ the problem of computing a Best Response is NP-hard and agents cannot afford exponential running time, they have to resort to an approximation of the best possible strategy. For that reason, we will analyze a greedy model in which the possible new strategies of a player are only those in which he adds, removes or swaps a link from his current strategy. In addition, we want to experiment to see how frequent the cycles are since there are previous results where cycles have been found. Furthermore, we will analyze the time of convergence, that is, the number of turns that we have to give to the players to reach an equilibrium.

1.3 Project outline

This document structure is the following one:

- In Chapter 2 we introduce the necessary definitions and concepts and we give a detailed description of the model to study.
- In Chapter 3 we study the Sum Celebrity Games dynamics for $\beta = 1$.
- In Chapter 4 we study the Max Celebrity Games dynamics for $\beta = 1$.
- In Chapter 5 we study the Sum and Max Celebrity Games dynamics for $\beta > 1$.
- In Chapter 6 we do the experimental study of Sum and Max Celebrity Games for $\beta > 1$.
- Finally, in Chapter 7 we summarize the main contributions and analyze the open problems.

2 Preliminaries

In this chapter we provide the basic concepts of Game Theory, we present the model of Celebrity Games, and we introduce the concept of dynamics. In addition, we summarize the previous known work about Celebrity Games.

2.1 Strategic Games

Since Network Creation Games and in particular Celebrity Games are Strategic Games, we first define the general concepts in order to later go into specific details.

Strategic Games model the interaction between decision-makers, that we usually refer to as *players*. Each player has a set of possible actions or strategies to choose and his decision may depend on the actions of the rest of the players.

Definition 1. *A strategic game is defined as a tuple* $\Gamma = \langle V, (\mathcal{S}(u))_{u \in V}, (c_u)_{u \in V} \rangle$ *where:*

- $V = \{1, ..., n\}$*, a set of players.*
- *for each player* $u \in V$*,* $\mathcal{S}(u)$ *is the set of strategies or actions.*

  The strategy profile $S = (S_1, ..., S_n)$ *is a n-tuple formed by the actions chosen by the players, where for each player* $i \in V$*,* $S_i \in \mathcal{S}(i)$*. Therefore, we denote all the possible strategy profiles of* Γ *as* $\mathcal{S}(\Gamma) = \mathcal{S}(1) \times ... \times \mathcal{S}(n)$*.*

  For a strategy profile $S = (S_{-u}, S_u) \in \mathcal{S}(\Gamma)$ *we denote by* (S_{-u}, S'_u) *the strategy profile that is obtained from* S *in which player* u *has changed his strategy* S_u *by* S'_u*,* $(S_{-u}, S'_u) = (S_1, ..., S_{u-1}, S'_u, S_{u+1}, ..., S_n)$*.*

- $c_u : \mathcal{S}(\Gamma) \to \mathbb{R}$*, is the cost function of player* u*.*

  It is worth noting that the value of c_u *may depend on the chosen actions or strategies of all players.*

  Given a strategy profile S *of a game, in order to compare a players cost depending on the chosen strategy we will denote by* $\Delta(S_{-u}, S'_u) = c_u(S_{-u}, S'_u) - c_u(S)$*.*

2.1.1 Nash Equilibria

An important concept related to Strategic Games is the concept called *Nash Equilibrium*, that we will refer as NE. In some games there is a strategy profile or configuration where no player wants to unilaterally modify his strategy since he cannot strictly decrease his current cost.

Definition 2. *Given a* strategic game $\Gamma = \langle V, (\mathcal{S}(u))_{u \in V}, (c_u)_{u \in V} \rangle$ *we say that a strategy profile* $S \in \mathcal{S}(\Gamma)$ *is a NE if*

$$\forall u \in V, \forall S'_u \in \mathcal{S}(u) \ \Delta(S_{-u}, S'_u) \geq 0$$

Since the agents act selfishly it is reasonable to think that they will choose their best strategy given the actions of the other players. For this purpose we introduce the concept of Best Response.

Definition 3. *Given a strategic game* $\Gamma = \langle V, (\mathcal{S}(u))_{u \in V}, (c_u)_{u \in V} \rangle$ *and* $S \in \mathcal{S}(\Gamma)$*, the set of Best responses of player* u *to* S_{-u} *is defined as follows:*

$$BR(S, u) = \{S^*_u \in \mathcal{S}(u) \mid \forall S'_u \in \mathcal{S}(u) \ c_u(S_{-u}, S^*_u) \leq c_u(S_{-u}, S'_u)\}$$

2.1.2 Price of Anarchy

In addition to the cost of each player we want to measure how good a configuration is for the whole society, that is, the set of players. In order to be able to evaluate this we define an objective function, usually known as *social cost*.

Definition 4. *Let* $\Gamma = \langle V, (\mathcal{S}(u))_{u \in V}, (c_u)_{u \in V} \rangle$ *be a strategic game. Given a strategy profile* $S \in \mathcal{S}(\Gamma)$ *we define the social cost as*

$$C(S) = \sum_{u \in V} c_u(S)$$

In this way, now we can compare configurations and know which is is socially better.

One way to evaluate whether the social cost of a configuration is good or not is to compare it against the optimal social cost. We define socially optimal configurations as follows.

Definition 5. *Let* $\Gamma = \langle V, (\mathcal{S}(u))_{u \in V}, (c_u)_{u \in V} \rangle$ *be a strategic game. A strategy profile* $S \in \mathcal{S}(\Gamma)$ *is socially optimal if*

$$\forall S' \in \mathcal{S}(\Gamma) \ C(S) \leq C(S')$$

Let $OPT(\Gamma)$ *be the set of all strategy profiles that are socially optimal in* Γ.

Note that strategy profiles with optimal social cost might not be NE.

A doubt that arises is: since agents act in a selfish way, how inefficient is the equilibrium for the fact of not cooperating for the benefit of the whole? It is considered that a configuration benefits the society if it minimizes the social cost.

To determine the quality of Nash equilibria we use the ratio between the worst social cost of the worst equilibrium and the optimum social cost. This ratio is know as *Price of Anarchy.*

Definition 6. *Let $\Gamma = \langle V, (\mathcal{S}(u))_{u\in V}, (c_u)_{u\in V}\rangle$ be a strategic game, let $NE(\Gamma) \subseteq \mathcal{S}(\Gamma)$ be the set of Nash Equilibria and $S^* \in OPT(\Gamma)$. Then the Price of Anarchy is*

$$PoA(\Gamma) = \frac{max_{S\in NE(\Gamma)} C(S)}{C(S^*)}$$

2.1.3 Example: The Prisoner's Dilemma

For better understanding we explain a classic example in order to clarify the different concepts introduced.

"Two crime suspects have to choose between confessing to a crime or stay quiet. They cannot interact with each other, so they cannot know beforehand the decision of the other. The possibilities are as follows: if both stay quiet they will not be sentenced to the maximum penalty but will be punished with 2 years. If one of them finks, his sentence will be reduced to 1 year and the other suspect will be punished with 5 years, that is the maximum penalty. If they both fink they will be sentenced to 4 years, instead of 5, for cooperating with the authorities."

Formally, the game is represented by $\Gamma = \langle V = \{1, 2\}, (S_1, S_2), (c_1, c_2)\rangle$, where

- The players of the game are the criminals.
- Their actions are identical since they can confess or stay quiet. $S_1 = S_2 = \{\text{Fink}, \text{Quiet}\}$.

  Therefore, $S = S_1 \times S_2 = \{(\text{Quiet}, \text{Quiet}), (\text{Quiet}, \text{Fink}), (\text{Fink}, \text{Quiet}), (\text{Fink}, \text{Fink})\}$
- We can represent the cost of each player as the years in prison:

$$\begin{aligned}
&c_1(\text{Quiet}, \text{Quiet}) = c_2(\text{Quiet}, \text{Quiet}) = 2\\
&c_1(\text{Fink}, \text{Fink}) = c_2(\text{Fink}, \text{Fink}) = 4\\
&c_1(\text{Quiet}, \text{Fink}) = c_2(\text{Fink}, \text{Quiet}) = 5\\
&c_1(\text{Fink}, \text{Quiet}) = c_2(\text{Quiet}, \text{Fink}) = 1
\end{aligned}$$

It can be represented in a compact way using the following table:

		Suspect 2	
		Quiet	Fink
Suspect 1	Quiet	2,2	5,1
	Fink	1,5	4,4

Table 2.1: Table of the cost incurred of the four configurations

We observe how the only situation of equilibrium is when both confess. In other cases, at least one player can improve his cost by going from not confessing to confessing. Even so, the social optimum occurs when both stay quiet.

In this case, because both do not seek the social optimum but instead act selfishly, we observe a PoA of $\frac{C(Fink,Fink)}{C(Quiet,Quiet)} = 2$.

In other words, the social cost of the worst equilibrium, which in this case is the only one, is twice the optimum social cost.

2.1.4 Example: A simple Network Creation Game

Let us introduce a closer example of what we will deal with throughout the project. We define the game as the tuple $\Gamma = \langle V, (w_u)_{u \in V}, \alpha \rangle$, where:

- $V = \{1, ..., n\}$ is the set of players.
- $w_u > 0$ is the weight of player $u \in V$.
- $\alpha > 0$ is the link cost.

Each player represents a node of a graph and the strategy of u is $S_u \in \mathcal{P}(V - \{u\})$, where $\mathcal{P}$ is the power set. The strategy of u represents the players to whom he buys a link.

Let the cost function of a player u be $c_u(S) = \alpha|S_u| + \sum_{\{v \mid v \notin S_u \wedge u \notin S_v\}} w_v$.

The strategy profile S determines an undirected graph, defined by $G[S] = (V, \{\{u, v\} \mid u \in S_v \vee v \in S_u\})$. The directed version of $G[S]$ also determines who have bought the link, (u, v) is an edge of the directed graph if $v \in S_u$.

Let $\Gamma = \langle \{0, 1, 2\}, (w_u)_{u \in V}, 4 \rangle$, where $w_0 = 7$, $w_1 = 6$ and $w_2 = 5$.

Let the strategy profile $S = (S_0, S_1, S_2) = (\emptyset, \{0\}, \emptyset)$, then we can represent the strategy profile as follows:

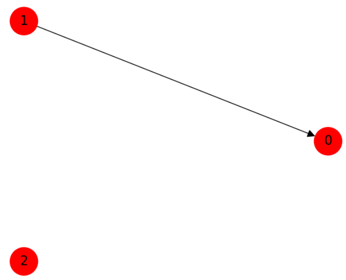


Figure 2.1: Strategy profile S of Γ

If we compute the Best Response for player 0 in S we have $S'_0 \in BR(S, 0)$. The new strategy profile is $S' = (S_{-0}, S'_0)$. It is easy to see that the Best Response for him is to buy the link to 2 since $w_2 > \alpha$ and is not yet adjacent.

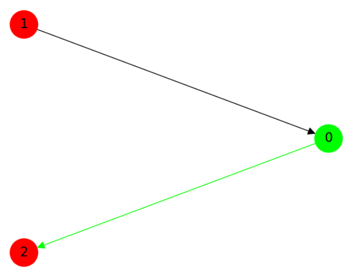


Figure 2.2: Strategy profile S' of Γ

If player 2 applies the Best Response to S' now he will buy a link to 0 for the same reason. At this time no player will want to change his strategy, so we will be in a situation of Nash equilibrium. It is easy to see that all the NE are a 3-clique in the undirected graph. Otherwise, if S is the strategy profile of Γ and u is a player, since $\forall i \in V\ \ w_i > \alpha$ if $v \notin S_u \wedge u \notin S_v$, we can define $S'_u = S_u \cup \{v\}$ and we have $\Delta(S_{-u}, S'_u) = \alpha - w_v < 0$.

2.2 Celebrity Games

Network Creation Games (NCG) are a type of strategic games that model how a network behaves based on the parameters that characterize it. By now the classic NCG model was proposed by Fabrikant et al. [1]. In this model agents without centralized coordination establish links to other nodes in order to be as well connected as possible in the network. The cost of each player has two components: firstly the cost of the purchased links and secondly the sum of the distance from the node to the rest.

In this project we are going to study Celebrity Games, a new model of Network Creation Games recently proposed by Àlvarez et al. [3, 4, 5]. This model allows us to characterize another point of view of the networks. First of all, this model assigns relevance to each agent through his weight. The heavier the weight, the more important he is in the network. Secondly, it includes the concept of critical distance. Instead of adding the distances between the node and the rest, it adds the weight of the nodes that are at a distance greater than the critical. In addition, it maintains the concept of link cost.

Intuitively, the goal of each agent in the celebrity games is to connect, or at least be at a reasonable distance, to the most relevant nodes by using the least number of links.

Definition 7. *A Celebrity Game is defined by* $\Gamma = \langle V, (w_u)_{u \in V}, \alpha, \beta \rangle$*, where:*

- $V = \{1, ..., n\}$ *is the set of players.*
- $w_u > 0$ *is the weight of player* $u \in V$*.*
- $\alpha > 0$ *is the link cost.*
- $1 \leq \beta \leq n - 1$ *is the critical distance.*

We use W to refer to the sum of the weights, $W = \sum_{u \in V}\{w_u\}$ as well as $w_{min} = min_{u \in V}\{w_u\}$ and $w_{max} = max_{u \in V}\{w_u\}$

A strategy for a player u in Γ is $S_u \subseteq V - \{u\}$, represents the set of nodes to which the player u has purchased a link. The strategy profile S of a game Γ is defined as in the strategic games $S = (S_1, ..., S_n)$.

Every strategy profile S determines a non-directed graph associated with it, defined by $G[S] = (V, \{\{u, v\} \mid u \in S_v \vee v \in S_u\})$. The directed version of $G[S]$ is given by S, i.e., (u, v) is an edge of the directed graph if $v \in S_u$.

The way we define the cost of player u in strategy profile S will bring us various variants of the game which we will study later. This versions are defined in [3, 4] and [5]. Their cost functions are respectively:

- *Sum Celebrity Game*: $c_u(S) = \alpha|S_u| + \sum_{\{v \mid d_{G[S]}(u,v) > \beta\}} w_v$
- *Max Celebrity Game*: $c_u(S) = \alpha|S_u| + \max_{\{v \mid d_{G[S]}(u,v) > \beta\}} w_v$

We will refer to the Sum and Max versions of Celebrity Game as SUM-CG and MAX-CG, respectively.

The following is a summary of the more relevant results in [3, 4, 5] for this project.

First of all, for both versions we have well differentiated results between $\beta = 1$ and $\beta > 1$.

In both models, for $\beta = 1$ the problem of computing a Best Response for a player is polynomial time solvable. In addition, in the case of SUM-CG we have that PoA is at most 2 and in the case of MAX-CG, $PoA = O(w_{max}/w_{min})$.

However, for $\beta > 1$ the problem of computing a Best Response for a player becomes NP-hard.

In addition, we have the following main results for the SUM-CG when $\beta > 1$:

- The optimal social cost of a game Γ depends on the relation between W and α, $OPT(\Gamma) = min\{\alpha, W\}(n-1)$.
- Every NE graph is either connected or the graph I_n.

  We refer to the game as star celebrity game if has a NE graph that is connected. Γ is star celebrity game if $\alpha < w_{max}$ or $\alpha \geq w_{max}$ and there is at most one $u \in V$ for which $\alpha > W - w_u$.
- If G is a NE graph of a star celebrity game then $diam(G) \leq 2\beta + 1$.
- If G is a NE graph of a star celebrity game then $PoA = O(min\{n/\beta, W/\alpha\})$.

And the main results for the MAX-CG when $\beta > 1$, are the following:

- The optimal social cost of a game Γ depends on the relation between w_{max} and α, $2\alpha(n-1) \geq OPT(\Gamma) \geq min\{\alpha(n-1), w_{max}(n-1) + w_{min}\}$.
- If G is a NE graph then $diam(G) \leq 2\beta + 2$.
- $PoA \leq 2(w_{max}/\alpha)$ and $PoA(\Gamma) = O(n/\beta)$.

As we can see, Celebrity Games have a different behavior for $\beta = 1$ and $\beta > 1$. For this reason we divide the theoretical part in each of the cases.

For $\beta = 1$ we make an in-depth study of the Best Response dynamics for both cases, since is the natural way the agents will make the changes and its computation is polynomial time solvable.

For $\beta > 1$, on the other hand, since the Best Response is NP-hard and agents cannot take exponential time to compute their response, we examine a greedy model where the response of a player is restricted. In this way, from a current strategy the only possible responses are those in which the player can add, delete or swap one link. In this way the problem of computing a Best Response is polynomial time. To conclude, Àlvarez &

Messegué show in [8] that the greedy model we analyze admits cycles for $\beta = 3$ and $\beta = 5$ for the Sum and Max models, respectively.

Formally, the greedy model is defined as follows:

Definition 8. *Greedy Celebrity Game (Greedy CG), agents can only: add, remove or swap one link. If the original strategy is S_u and the new is S'_u then: if added new edge $S_u \subset S'_u$ and $|S_u| = |S'_u| - 1$, if removed an edge $S_u \supset S'_u$ and $|S_u| - 1 = |S'_u|$ and finally, if swapped $|S_u| = |S'_u|$ and $|S_u \cap S'_u| = |S_u| - 1$.*

This model induces a weaker equilibrium configuration:

Definition 9. *Given a celebrity game $\Gamma = \langle V, (w_u)_{u \in V}, \alpha, \beta \rangle$ we say that a strategy profile $S \in \mathcal{S}(\Gamma)$ is a Greedy Equilibrium (GE) if no agent can unilaterally strictly decrease her cost by either buying or deleting or swapping one own edge.*

In comparison with Nash Equilibrium, the Greedy Equilibrium is clearly a much weaker solution. We have that $NE \subseteq GE$ since if no agent can change his strategy to improve the cost, then certainly no agent can improve by buying, deleting or swapping one own edge.

2.3 Dynamics in Celebrity Games

In order to analyze the dynamic behavior of networks whose topology and quality of equilibra depends on the critical distance, the price per link and the weight of the different nodes, we study the dynamics of Sum and Max Celebrity Games, as well as both versions of Greedy Celebrity Games.

We consider that the dynamic process is divided into rounds, in each round is selected a single player who can change his strategy whenever it improves his current cost. How the rounds are distributed among the players and how the change of strategy is chosen determines the dynamics. We denote by $S^{(i)}$ the strategy profile or configuration at round $i \geq 0$ and $S_u^{(i)}$ the strategy of player u in $S^{(i)}$.

Since only one agent can change his strategy at every round we need a *turn policy* that determines how the rounds are distributed among the players. We call *response policy* the way in which a player determines his new strategy. At each round $i \geq 0$, $S^{(i)}$ is updated to $S^{(i+1)}$ by changing the strategy of a player u, selected by the turn policy, from $S_u^{(i)}$ to $S_u^{(i+1)}$ by applying the response policy.

In order to define completely a network creation process we combine the following ingredients: the version of the game (one of the four different combinations between SUM and MAX with CG and Greedy CG), an initial configuration, the response policy and a turn policy. If some of these characteristics are not stated is because it is arbitrary. We will often indicate the response policy before the name of the dynamics.

3 Dynamics in Sum Celebrity Games for $\beta = 1$

In this section we first analyze the computational cost of deciding whether a strategy profile is NE or not. Subsequently, we analyze the Best Response SUM-CG dynamics in order to determine if cycles are possible and if there is a dynamics that reach an equilibrium in a reasonable computational time.

By Proposition 10 of [4] we have the following result:

Proposition 1. *The problem of computing a Best Response of a player u to a strategy profile in SUM-CG is polynomial time solvable.*

Corollary 1. *The problem of deciding whether a given strategy profile is a NE in SUM-CG is polynomial time computable.*

Proof. By definition we have that $S = (S_1, ..., S_n)$ is NE if

$$\forall u \in V, \forall S'_u \in \mathcal{S}(u) \; \Delta(S_{-u}, S'_u) \geq 0$$

Since $\forall u \in V \; \forall S'_u \in \mathcal{S}(u) \; c_u(S_{-u}, S'_u) \geq c_u(S_{-u}, S^*_u)$ where $S^*_u \in BR(S, u)$ we only have to check that

$$\forall u \in V \; \Delta(S_{-u}, S^*_u) \geq 0$$

Then, given S, the algorithm goes through all the players and checks that for each player his Best Response it is not better than their current strategy. Since executing the BR is polynomial time computable, the overall execution of $|V|$ times BR is also polynomial time. □

Now let us show some basic properties of SUM-CG when $\beta = 1$ that will be useful to us later.

Let $V_{>\alpha} = \{u \in V \mid w_u > \alpha\}$, $V_{=\alpha} = \{u \in V \mid w_u = \alpha\}$ and $V_{<\alpha} = \{u \in V \mid w_u < \alpha\}$.

Property 1. *Let $\Gamma = \langle V, (w_u)_{u \in V}, \alpha, 1\rangle$, $S \in \mathcal{S}(\Gamma)$, $S'_u \in BR(S, u)$ and $S' = (S_{-u}, S'_u)$.*

For an arbitrary player u we analyze for any $v \neq u$ whether $v \in S'_u$ or not:

(a) If $u \in S_v$ then $v \notin S'_u$.

Suppose $v \in S'_u$, define $S''_u = S'_u - \{v\}$. We have $\Delta(S'_{-u}, S''_u) = -\alpha < 0$, which contradicts $S'_u \in BR(S, u)$.

(b) If $u \notin S_v$, we have the following three cases:

(1) $v \in V_{>\alpha} \implies v \in S'_u$.

Suppose $v \notin S'_u$, define $S''_u = S'_u \cup \{v\}$. We have $\Delta(S'_{-u}, S''_u) = \alpha - w_v < 0$, which contradicts $S'_u \in BR(S, u)$.

(2) $v \in V_{<\alpha} \implies v \notin S'_u$.

Suppose $v \in S'_u$, define $S''_u = S'_u - \{v\}$. We have $\Delta(S'_{-u}, S''_u) = w_v - \alpha < 0$, which contradicts $S'_u \in BR(S, u)$.

(3) $v \in V_{=\alpha} \implies (v \in S'_u \vee v \notin S'_u)$.

The cost of buying or not the link is the same, since $w_v = \alpha$ then both situations are possible.

Proposition 2. *For any initial configuration and any turn policy the Best Response SUM-CG dynamics has no cycles.*

Proof. Assume for contradiction that a cycle exists. By definition a cycle exists $\iff$ $\exists i, j, k$ with $i < j < k : S^{(i)} \neq S^{(j)} \wedge S^{(i)} = S^{(k)}$. Therefore, a cycle exists if $\exists i, j, k$ with $i < j < k$ and $\exists u, v \in V$ $((v \notin S^{(i)}_u \wedge v \in S^{(j)}_u \wedge v \notin S^{(k)}_u) \vee (v \in S^{(i)}_u \wedge v \notin S^{(j)}_u \wedge v \in S^{(k)}_u))$

Hence, there is at least a node u such that alternates between having a link to v in his strategy and deleting such link. Such agent with $w_v \neq \alpha$ exists, otherwise the cost would remain the same by Property 1.(b.3), and by definition the cost has to be strictly decreasing.

If u first does not have the link, the only possibility to buy the link is to apply Property 1.(b.1), since it is the only one that as a result u buys the link when $w_v \neq \alpha$. Therefore, $w_v > \alpha$. We have assumed u deletes this link in a subsequent round, but by Property 1.(a) the node v will not buy a link to u and as a result by Property 1.(b.1) the node u will not delete the link.

The argument is analogous if initially u has the link but applying the properties in the opposite order.

This contradicts our assumption that there is a cycle when using Best Response as response policy. $\square$

Proposition 3. *The Best Response SUM-CG dynamics with a Round-robin turn policy converges to a NE in at most $2|V|$ rounds.*

Proof. Let $\Gamma = \langle V, (w_u)_{u \in V}, \alpha, 1 \rangle$ be a SUM-CG and $S^{(0)}$ any initial configuration.

The dynamics will consist of a sequence $S^{(0)}, ..., S^{(2n)}$ where in the transition from $S^{(i)}$ to $S^{(i+1)}$ the $((i \mod n) + 1)$-th node updates its strategy applying the Best Response.

The sequence $S^{(0)}, ..., S^{(2n)}$ converges to a NE.

Let $V_{\neq\alpha} = \{u \in V \mid w_u \neq \alpha\}$ and let $u, v \in V_{\neq\alpha}$ two arbitrary agents such that $u \neq v$. We first analyze the existence or non-existence of edges between two given agents after the $2n$ rounds depending on their weights:

(a) $u, v \in V_{>\alpha}$.

In the first n rounds we have that $u \notin S_v^{(n)} \vee v \notin S_u^{(n)}$, since at least one node would apply Property 1.(a). But at the very least, there will be a node that has purchased the link by Property 1.(b.1).

In n rounds the existence of the edge is definitive.

(b) $u, v \in V_{<\alpha}$.

In the first round of each node they remove (or not buy) the link, either by Property 1.(a) or 1.(b.2).

In n rounds the non-existence of the edge is definitive.

(c) $u \in V_{>\alpha}$ and $v \in V_{<\alpha}$.

In the first n rounds u will delete (or not buy) the link to v by Property 1.(a) or 1.(b.2).

In the following n rounds the node v will buy the link (or not delete) by Property 1.(b.1).

In $2n$ rounds the existence of the edge is definitive.

In the worst case we need $2n$ rounds in order to guarantee the existence or non-existence of a link between two nodes of weight different than α. Let us prove that $S^{(2n)}$ is a NE.

Suppose that $S^{(2n)}$ is not a NE. Therefore, there must be a node that can improve its cost, let this node be $u \in V$.

Let us refer to $2n$ as k and let $S_u^{(k+1)} \in BR(S^{(k)}, u)$, $S^{(k+1)} = (S_{-u}^{(k)}, S_u^{(k+1)})$ and let $A = S_u^{(k+1)} \setminus S_u^{(k)}$, $D = S_u^{(k)} \setminus S_u^{(k+1)}$. As we have seen, the edges between nodes with weight other than α are definitive, for that reason, the elements of $A \cup D$ have weight equal to α.

As u can improve its cost we have $\Delta(S_{-u}^{(k)}, S_u^{(k+1)}) < 0$.

$$
\begin{aligned}
&\Delta(S_{-u}^{(k)}, S_u^{(k+1)}) = \\
&= c_u(S^{(k+1)}) - c_u(S^{(k)}) = \\
&= \alpha|S_u^{(k+1)}| + \sum_{\{v \mid d_{G[S^{(k+1)}]}(u,v)>1\}} w_v - \left[\alpha|S_u^{(k)}| + \sum_{\{v \mid d_{G[S^{(k)}]}(u,v)>1\}} w_v\right] = \\
&= \alpha(|S_u^{(k+1)}| - |S_u^{(k)}|) + \sum_{\{v \mid d_{G[S^{(k+1)}]}(u,v)>1\}} w_v - \sum_{\{v \mid d_{G[S^{(k)}]}(u,v)>1\}} w_v =
\end{aligned}
$$

$$= \alpha(|A| - |D|) + \sum_{\{v \neq u \mid v \notin S_u^{(k+1)} \wedge u \notin S_v^{(k+1)}\}} w_v - \sum_{\{v \neq u \mid v \notin S_u^{(k)} \wedge u \notin S_v^{(k)}\}} w_v$$

We have that $V - \{u\} = S_u^{(k)} \cup \{v \neq u \mid u \in S_v^{(k)}\} \cup \{v \neq u \mid v \notin S_u^{(k)} \wedge u \notin S_v^{(k)}\}$. Intuitively, these sets are: the nodes to which u has purchased a link, the nodes that have purchased a link to u and the nodes that are not directly connected to u in round k. These sets are disjoint, since there is no connection purchased by both endpoints at round $2n$, and are as a result a partition of $V - \{u\}$. The same applies for $k + 1$.

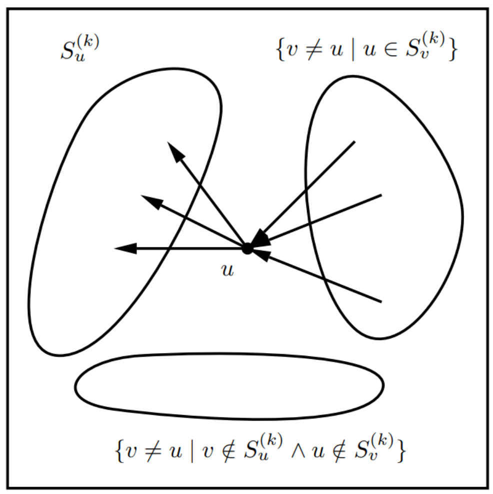


Figure 3.1: Partition of $V - \{u\}$ in round k

$$\begin{aligned}
&\alpha(|A| - |D|) + \sum_{\{v \neq u \mid v \notin S_u^{(k+1)} \wedge u \notin S_v^{(k+1)}\}} w_v - \sum_{\{v \neq u \mid v \notin S_u^{(k)} \wedge u \notin S_v^{(k)}\}} w_v = \\
&= \alpha(|A| - |D|) + \sum_{\{v \neq u \mid v \notin S_u^{(k+1)}\}} w_v - \sum_{\{v \neq u \mid u \in S_v^{(k+1)}\}} w_v \\
&\qquad - \left[\sum_{\{v \neq u \mid v \notin S_u^{(k)}\}} w_v - \sum_{\{v \neq u \mid u \in S_v^{(k)}\}} w_v \right]
\end{aligned}$$

Since $S_v^{(k)} = S_v^{(k+1)}$ when $v \neq u$,

$$\begin{aligned}
&\alpha(|A| - |D|) + \sum_{\{v \neq u \mid v \notin S_u^{(k+1)}\}} w_v - \sum_{\{v \neq u \mid u \in S_v^{(k+1)}\}} w_v \\
&\qquad - \left[\sum_{\{v \neq u \mid v \notin S_u^{(k)}\}} w_v - \sum_{\{v \neq u \mid u \in S_v^{(k)}\}} w_v \right] =
\end{aligned}$$

$$= \alpha(|A| - |D|) + \sum_{\{v \neq u \mid v \notin S_u^{(k+1)}\}} w_v - \sum_{\{v \neq u \mid v \notin S_u^{(k)}\}} w_v$$

Since $D = S_u^{(k)} \setminus S_u^{(k+1)}$, $A = S_u^{(k+1)} \setminus S_u^{(k)}$ and $\forall v \in (A \cup D)\ w_v = \alpha$,

$$\begin{aligned}
&\alpha(|A| - |D|) + \sum_{\{v \neq u \mid v \notin S_u^{(k+1)}\}} w_v - \sum_{\{v \neq u \mid v \notin S_u^{(k)}\}} w_v = \\
&= \alpha(|A| - |D|) + \sum_{v \in D} w_v - \sum_{v \in A} w_v = \\
&= \alpha(|A| - |D|) + \alpha|D| - \alpha|A| = 0
\end{aligned}$$

As a result, contradiction with the fact that $S^{(2n)}$ is not a NE.

□

Note that there are configurations in which we only need n rounds. For example, when the induced graph of the initial network is the null graph.

By Proposition 3, starting from any strategy profile of a SUM-CG it converges to a NE. Therefore, there is always at least one NE.

To obtain a NE we need at most $2|V|$ rounds, where in each round its executed a BR. Since executing the BR is polynomial the overall execution $2|V|$ times BR is also polynomial time.

Corollary 2. *There always exists a NE of a given SUM-CG.*

Corollary 3. *The problem of computing a NE of a given SUM-CG is polynomial time solvable.*

4 Dynamics in Max Celebrity Games for $\beta = 1$

In this section we are going to study the same questions as in the case of Sum. That is to say, the computational time to decide if a strategy profile is NE and determine if cycles are possible, together with the study of defining a dynamics that reaches NE in reasonable computation time.

By Proposition 10 of [5] we have the following result:

Proposition 4. *The problem of computing a Best Response of a player u to a strategy profile in MAX-CG is polynomial time solvable.*

Corollary 4. *The problem of deciding whether a given strategy profile is a NE in MAX-CG is polynomial time computable.*

The proof of the previous corollary is analogous to Corollary 1.

In this case we are going to proof that the Sum and Max models have a different behavior, since in this last case cycles are possible in the Best Response dynamics.

Proposition 5. *There exists an initial configuration and a turn policy for which the Best Response MAX-CG dynamics cycles when $|V| \geq 3$.*

Proof. We give an example where this situation occurs. Since each strategy profile uniquely determines a directed graph where each arc (u, v) indicates that u has bought a link to v, we use this form to represent the cycle of the dynamics. The initial graph has three nodes, $V = \{0, 1, 2\}$ with $w_0 = 7$, $w_1 = 6$ and $w_2 = 5$. In addition, $\alpha = 4$.

Initially we have $S_1^{(0)} = \{0\}$ and $S_0^{(0)} = S_2^{(0)} = \emptyset$.

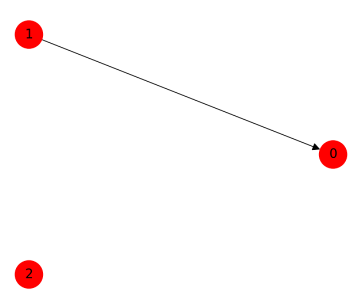
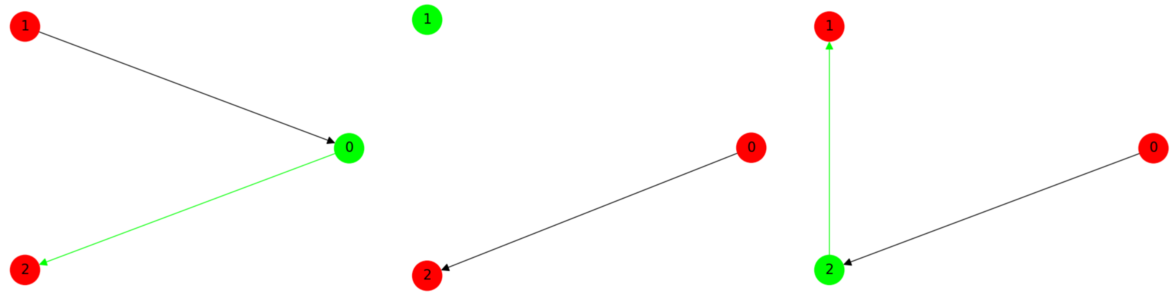


Figure 4.1: Initial configuration, $S^{(0)}$

Because the example has few nodes it is easy to verify that each step is the a Best Response. The turn policy selects the agents in decreasing order of weight. Therefore, the order of the nodes will be: $0, 1, 2, 0, ...$

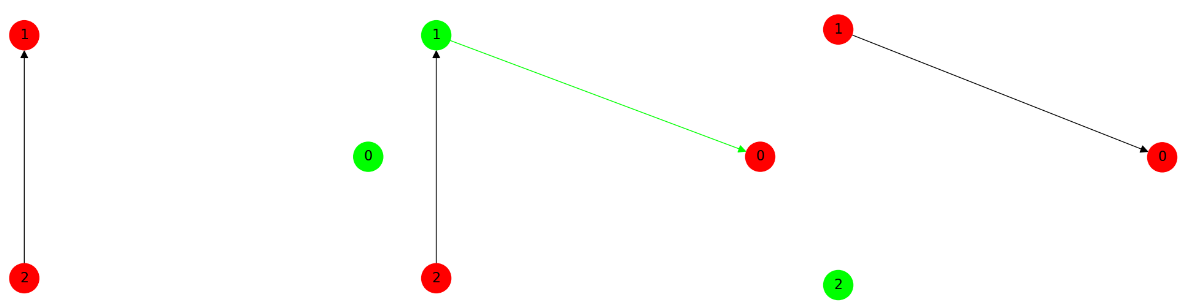


$S^{(1)}$ $S^{(2)}$ $S^{(3)}$

Figure 4.2: First three steps

- *Step 0-1*: given that node 1 has bought an edge to node 0 it can improve the cost by buying the remaining edge. It will decrease the cost from $c_0(S^{(0)}) = w_2 = 5$ to $c_0(S^{(1)}) = \alpha = 4$.
- *Step 1-2*: node 1 deletes its initial edge. The cost goes from $c_1(S^{(1)}) = w_2 + \alpha = 5 + 4 = 9$ to $c_1(S^{(2)}) = w_0 = 7$.
- *Step 2-3*: a situation similar to the first one occurs. Node 2 takes advantage of the fact that buying the remaining edge can reduce the cost from $c_2(S^{(2)}) = w_1 = 6$ to $c_2(S^{(3)}) = \alpha = 4$.

As we can see, there is a cyclic process in which a node buys a link since it only has a non-adjacent node and then the node that initially had the link deletes it.



$S^{(4)}$ $S^{(5)}$ $S^{(6)} = S^{(0)}$

Figure 4.3: Last three steps

This pattern repeats itself and after $S^{(6)}$ the next round will be for the node 0, producing as a result $S^{(1)}$ and therefore, a cycle.

We have seen an example for the specific case of $|V| = 3$, let us generalize the structure of the initial configuration to a configuration with $K > 3$ nodes.

Consider the game $\Gamma =< V, (w_u)_{u \in V}, \alpha, 1 >$, where

- $V = \{0, 1, 2\} \cup \{a_1, ..., a_{K-3}\}$, where $\{0, 1, 2\}$ are the nodes used before and a_i are the auxiliary nodes in order to have $|V| = K$.
- Define w_i for $i \in \{0, 1, 2\}$ as before and $w_{a_j} = \frac{1}{2}$ for $j \in \{1, ..., K-3\}$.
- $\alpha = 4$.

As none of the nodes $i \in \{0, 1, 2\}$ have incentive to buy a link to a_j with $j \in \{1, ..., K-3\}$, we obtain obtain a cycle when the turn policy is $0, 1, 2, 0, ...$

This cause the cycle to continue to occur for $|V| \geq 3$. $\square$

Proposition 6. $\forall i \geq 0, \forall u \in V$ $S_u^{(i)} \subseteq S_u^{(i+1)}$ *in the Best Response MAX-CG dynamics when* $S^{(0)} = (\emptyset, ..., \emptyset)$.

Proof. We use strong induction on $i \geq 0$. The base case is true since for $i = 0$ and an arbitrary node $u \in V$ we have $S_u^{(0)} = \emptyset$, and therefore $\emptyset = S_u^{(0)} \subseteq S_u^{(1)}$.

Inductive hypothesis: Let $k \geq 0$ and assume that for j, $0 \leq j \leq k$ it holds $\forall u \in V : S_u^{(j)} \subseteq S_u^{(j+1)}$.

Let us show that for any arbitrary $u \in V : S_u^{(k+1)} \subseteq S_u^{(k+2)}$.

If $S_u^{(k+1)} = \emptyset \implies \emptyset = S_u^{(k+1)} \subseteq S_u^{(k+2)}$ as in the base case.

If at round $k+1$ is not the turn of player u then $S_u^{(k+1)} = S_u^{(k+2)}$ which trivially implies $S_u^{(k+1)} \subseteq S_u^{(k+2)}$.

Otherwise, $\exists j \in \{0, ..., k\}$ such that at round j the player u updated his strategy to $S_u^{(j+1)}$. Let j be the last round that the node changed its strategy, in this way we have $S_u^{(k+1)} = S_u^{(j+1)}$.

By Proposition 10 of [5] we know that the Best Response of a player u at round t can be computed by defining $Y_u^{(t)} = \{v \neq u \mid u \notin S_v^{(t)}\}$ and assuming w.l.o.g $Y_u^{(t)} = \{y_1^{(t)}, ..., y_{r_t}^{(t)}\}$ with $w_{y_1^{(t)}} \geq ... \geq w_{y_{r_t}^{(t)}}$. Then, for each $l \leq r_t$, the strategy of cardinality l that minimize c_u is $\{y_1^{(t)}, ..., y_l^{(t)}\}$. Finally, the Best Response can be computed by finding the $l \in \{0, ..., r_t\}$ such that minimize $\alpha l + w_{y_{l+1}^{(t)}}$, assuming $w_{y_{r_t+1}}^{(t)} = 0$. We make this assumption because node u would have purchased all non-adjacent nodes and therefore there will be no nodes at a greater distance than the critical distance.

We define a canonical order in order to ensure that we always get the same result in case that there is more than one possible Best Response. In the case of a weight tie, we select by increasing order of label, that is $y_i^{(t)} < y_{i+1}^{(t)}$ if $w_{y_i^{(t)}} = w_{y_{i+1}^{(t)}}$ $\forall i \in \{1, ..., r_t - 1\}$ and in the case of multiple possible l, we choose the smallest one.

By inductive hypothesis we have

$$Y_u^{(j)} = \{v \neq u \mid u \notin S_v^{(j)}\} \supseteq \{v \neq u \mid u \notin S_v^{(k+1)}\} = Y_u^{(k+1)} \text{ , since } j < k+1$$

In addition, as no node $v \in S_u^{(j+1)}$ will buy a link to u in rounds $j+1$,...,k, we have that $Y_u^{(k+1)} \supseteq S_u^{(j+1)}$ since this edge already exists, and therefore buying a link would not be Best Response.

At any iteration t the Best Response uses the canonical order to sort the nodes from $Y_u^{(t)}$ and selects its first $|S_u^{(t+1)}|$ consecutive nodes in increasing order as the new strategy for player u. Using that $Y_u^{(j)} \supseteq Y_u^{(k+1)} \supseteq S_u^{(j+1)}$ and defining $q = |S_u^{(j+1)}|$ we have,

$$y_1^{(j)} = y_1^{(k+1)}, ..., y_q^{(j)} = y_q^{(k+1)} \tag{4.1}$$

In other words, the first q nodes in $Y_u^{(j)}$ are also the first in $Y_u^{(k+1)}$, applying in both situations the canonical order.

We consider several cases depending on the value of $l = |S_u^{(k+2)}|$.

1. $l \geq q$. The Best Response in round $k+1$ selects at least the first q nodes of $Y_u^{(k+1)}$ in the canonical order. As the first q elements are precisely the elements of $S_u^{(j+1)}$ we have $S_u^{(k+2)} \supseteq S_u^{(j+1)} = S_u^{(k+1)}$

2. $l < q$. Since the response policy of the node u at round j is a Best Response and it chooses the smallest cardinal set in case of a tie, we have that

   $$\alpha l + w_{y_{l+1}^{(j)}} > \alpha q + w_{y_{q+1}^{(j)}} \tag{4.2}$$

   In the round $k+1$ the node u has selected as Best Response the set of nodes $S_u^{(k+2)}$ so that the cost of player u now becomes $\alpha l + w_{y_{l+1}^{(k+1)}}$.

   By Best Response and canonical order we have,

   $$\alpha l + w_{y_{l+1}^{(k+1)}} \leq \alpha q + w_{y_{q+1}^{(k+1)}}$$

   By (4.1) we have $y_{l+1}^{(j)} = y_{l+1}^{(k+1)}$ and by $Y_u^{(j)} \supseteq Y_u^{(k+1)}$ and canonical order we have $w_{y_{q+1}^{(j)}} \geq w_{y_{q+1}^{(k+1)}}$.

   Therefore, $\alpha l + w_{y_{l+1}^{(j)}} = \alpha l + w_{y_{l+1}^{(k+1)}} \leq \alpha q + w_{y_{q+1}^{(k+1)}} \leq \alpha q + w_{y_{q+1}^{(j)}}$. Contradiction with expression (4.2).

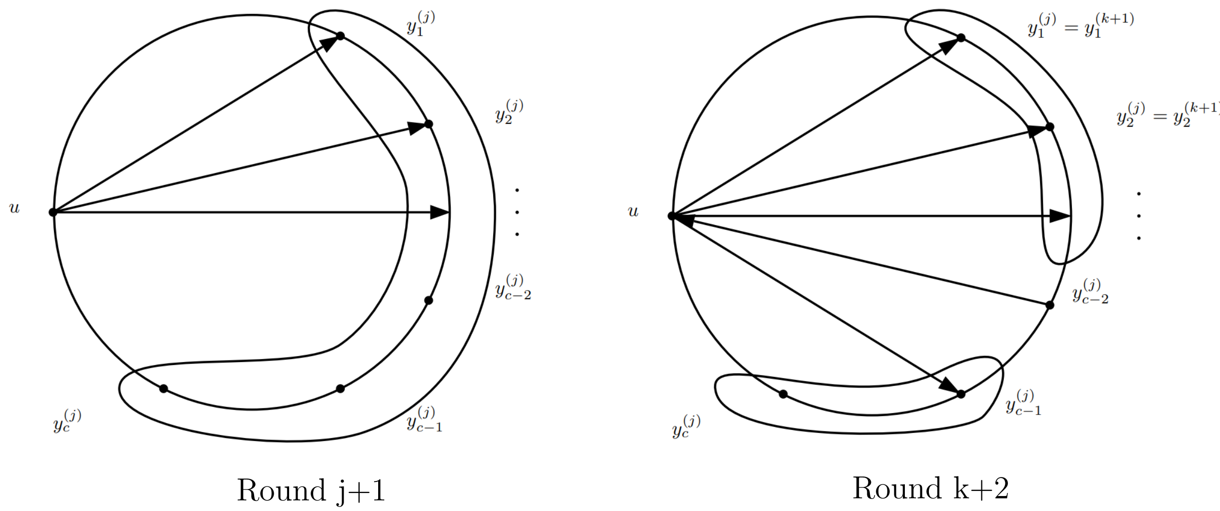


Figure 4.4: Visual representation of $Y^{(j)}$, $Y^{(k+1)}$ and the respective Best Responses

□

Although the Best Response MAX-CG dynamics allows cycles we are going to see that we can obtain a NE with a reasonable computational time.

Corollary 5. *Best Response MAX-CG dynamics and turn policy by enumeration of nodes converges to a NE in $O(|V|^3)$ rounds when $S^{(0)} = (\emptyset, ..., \emptyset)$.*

Proof. In the worst case, after $|V|$ rounds the number of edges is increased by 1. There are $O(|V|^2)$ edges and we need $|V|$ rounds for each. Therefore, $O(|V|^3)$ rounds in total. As the BR is polynomial time solvable the overall execution is also polynomial time. □

Corollary 6. *There always exists a NE of a given MAX-CG.*

Corollary 7. *The problem of computing a NE of a given MAX-CG is polynomial time solvable.*

5 Dynamics in Celebrity Games for $\beta > 1$

Since the problem of computing a Best Response for β >1 in Sum and Max is NP-hard we are going to study if we can obtain cycles by applying a Better Response policy for any β and α. This response policy is much more flexible, but it might also be the case that we have to examine an exponential number of configurations.

Prior to this result, Àlvarez & Messegué showed in [8] that the Better Response dynamics admits cycles for $\beta = 3$ and $\beta = 5$ for the Sum and Max models, respectively. In this case we are going to generalize for any $\beta > 1$ and any $\alpha > 0$.

Proposition 7. $\forall\beta > 1, \forall\alpha > 0$ *there exists an initial configuration and a turn policy for which the Better Response SUM-CG dynamics have cycles.*

Proof. We give a network which induces a Better Response cycle for any $\beta > 1$ and any $\alpha > 0$. Together with the network, we provide the inequalities that have to be satisfied in order to produce the cycle. Finally, we provide a solution to the inequalities where weights are a function of an arbitrary α.

Let $\Gamma = \langle\{0, 1, 2, 3\}, (w_u)_{u\in V}, \alpha, \beta\rangle$ be the SUM-CG and let the turn policy be: 2, 3, 2, 3, ...

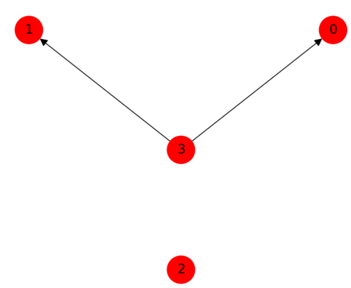
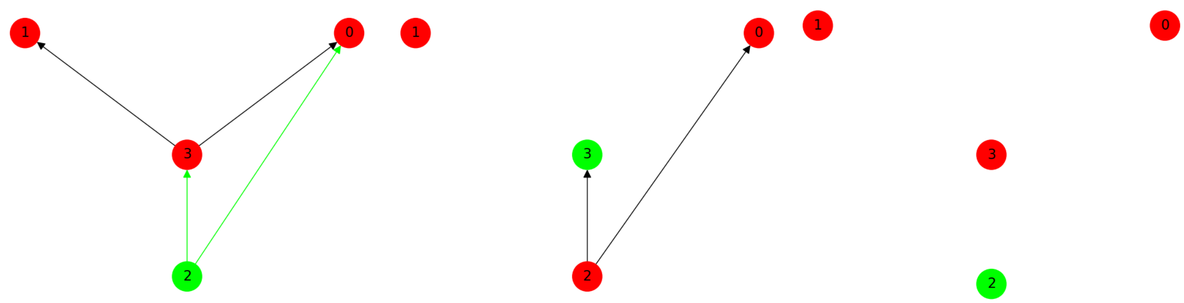


Figure 5.1: Initial network, $S^{(0)}$



$S^{(1)}$ $S^{(2)}$ $S^{(3)}$

Figure 5.2: The steps of a Better Response cycle for SUM, $S^{(4)} = S^{(0)}$

- *Step 0-1*: $c_2(S^{(0)}) = w_0 + w_1 + w_3 > 2\alpha = c_2(S^{(1)})$. Initially, the cost of node 2 is the sum of the weights of the rest of nodes, since is completely isolated. When node 2 buys a link to node 3 and 0 then it only has to pay the cost of the links, as it has all nodes at distance at most 2. Since $\beta > 1$ the distances does not exceed β.
- *Step 1-2*: $c_3(S^{(1)}) = 2\alpha > w_1 = c_3(S^{(2)})$. Node 3 delete the links and goes from having a cost of 2α to only pay the weight of node 1, since $\beta > 1$ and $d_{G[S^{(2)}]}(3, 0) = 2$.
- *Step 2-3*: $2\alpha + w_1 > w_0 + w_1 + w_3 \iff c_2(S^{(2)}) = 2\alpha > w_0 + w_3 = c_2(S^{(3)})$. Node 2 goes from paying the cost of two links and the weight of node 1 to pay the weight of the rest of nodes.
- *Step 3-4*: $w_0 + w_1 + w_2 > 2\alpha + w_2 \iff c_3(S^{(3)}) = w_0 + w_1 > 2\alpha = c_3(S^{(4)})$. Node 3 goes from paying the weight of the rest of nodes to pay the weight of node 2 and two links.

Let us note that the $\beta > 1$ parameter has been taken into account for the calculation of each of the costs. Since either the nodes are disconnected or if they are in the same connected component the distance is at most 2.

A possible solution for an arbitrary α:

$$w_0 = \frac{3}{2}\alpha,\ w_1 = \alpha,\ w_2 = \alpha,\ w_3 = \frac{1}{3}\alpha$$

□

Proposition 8. $\forall \beta > 1, \forall \alpha > 0$ *there exists an initial configuration and a turn policy for which the Better Response MAX-CG dynamics have cycles.*

Proof. We follow the same procedure used in the Proposition 7, in addition to use the same example.

Let $\Gamma = \langle \{0, 1, 2, 3\}, (w_u)_{u \in V}, \alpha, \beta \rangle$ be the MAX-CG. Initially we consider that $w_0 > w_1 > w_2 > w_3$. Let the turn policy be: 2, 3, 2, 3, ...

The reasoning is analogous to the one used before but in this taking into account that we are in MAX-CG. Instead of computing the sum of the weight we compute the max.

- *Step 0-1*: $c_2(S^{(0)}) = w_0 > 2\alpha = c_2(S^{(1)})$.
- *Step 1-2*: $c_3(S^{(1)}) = 2\alpha > w_1 = c_3(S^{(2)})$.
- *Step 2-3*: $c_2(S^{(2)}) = 2\alpha + w_1 > w_0 = c_2(S^{(3)})$.
- *Step 3-0*: $c_3(S^{(3)}) = w_0 > 2\alpha + w_2 = c_3(S^{(4)})$.

As before, let us note that the $\beta > 1$ parameter has been taken into account for the calculation of each of the costs.

A possible solution for an arbitrary α:

$$w_0 = 3\alpha,\ w_1 = \frac{3}{2}\alpha,\ w_2 = \frac{1}{2}\alpha,\ w_3 = \alpha$$

$\square$

We have observed that in both cases Better Response cycles are possible.

6 Experimental Results for $\beta > 1$

6.1 Experimental Setup

It is well-known that computing a Best Response of Celebrity Games when $\beta \geq 2$ is NP-hard. Since agents cannot afford exponential time to perfom the change of strategy, we are going to explore the greedy model in which a Best Response is polynomial time solvable. The Best Response in this case will consist in analyzing the best edge-addition, edge-swap and edge-delete. Then, the strategy is updated if with one of these changes we obtain a lower cost. In the case of ties, we prioritize the delete, then the swap and finally the addition.

We consider two turn policies: the *random policy*, where the node is selected randomly in each round, and the *max cost policy*, in which we are giving the turn in decreasing order of cost.

We consider two possible initial graphs: depending on whether the number of edges is fixed or not. In both cases we do not allow multi-edges. The generation of the graph in the first case consists of crossing all the combinations of two nodes without repetition and then randomly deciding the existence of the edge and its owner. In the second case, we select two nodes randomly and check whether they are different and there is no edge between them, in this case we add it. We repeat this process until m edges have been added.

The study of the impact of the initial topology is made through three types of topologies: *random*, *rl* (random line) and *dl* (directed line). In the *random*, we generate a graph with n vertices and $n-1$ edges. In the *rl*, we first generate a path of n nodes and then we choose the owner of each edge in a random way. On the other hand, in the *dl* the path is generated in the same way but the directions are chosen so that all edges point in the same direction.

For each one of the parameter configurations we run a total of 50 executions, unless otherwise indicated.

In each experiment we analyze:

- Average and maximum steps until converge. We refer to these functions as avg-steps and max-steps, respectively.
- Diameters, comparing with the theoretical results of NE in Celebrity Games.

- Maximum ratio between social cost and optimal social cost of the Greedy Equilibrium. We refer as r_{SUM-GE} and r_{MAX-GE} to that ratio of the Sum Greedy CG and Max Greedy CG, respectively. Each of these ratios is compared with the PoA of the respective model in Celebrity Games.

6.2 Experimental study of Dynamics in Sum Greedy Celebrity Games

It is natural to think of experimenting with weights as a function of α. We are going to experiment with different weight distributions that we consider to be representative.

In our experiments we consider the following scenarios:

- $n \leq 35$ when the weights are the same and $n \leq 25$, otherwise. The reason is computational limits.
-
  - All weights are identical w and are related to α as follows: $w = k\alpha$ for $k = 1$, $k < 1$ and $k > 1$. In this way we explore the different possibilities. We use $k = 20$ since Àlvarez et al. [3] showed that if there are more than one node u such that $\alpha > w(n-1)$ then I_n was the only NE, otherwise ST_n was NE of Γ.
  - The other possibility is that there are different types of weights:: T_1, T_2, T_3 or T_4, where:
    - T_1: There is one node more important than the rest. We use as weight of the important node $w = 20\alpha$ and for the rest $w = \alpha$.
    - T_2: There are three types of nodes with approximately the same amount in each. The three possible weights are: $w = 20\alpha$, $w = \alpha$ and $w = \alpha/20$.
    - T_3: The weights are distributed equidistant between $w_{min} = \alpha/20$ and $w_{max} = \alpha$.
    - T_4: The weights are distributed equidistant between $w_{min} = \alpha/20$ and $w_{max} = 20\alpha$.

    With T_3 and T_4 we will analyze the impact of the distance between w_{min} and w_{max}.
- β parameter: $\beta = 2$ or $\beta > 2$ with: $\beta = n/2$, $\beta = n/3.5$ and $\beta = n/5$. We choose these possibilities because they are the most representative for the n we test.
- Initial edge density: $m = 0$, $m = n$ and $m = 4n$.
- Initial topology: *random*, *rl* or *dl*.

As the weights we are going to study are in function of α we can prove the following property.

Proposition 9. *Let* $\Gamma = \langle V, (w_u)_{u\in V}, \alpha, \beta\rangle$ *and* $\Gamma' = \langle V, (w_u/\lambda)_{u\in V}, \alpha/\lambda, \beta\rangle$ *with* $\lambda > 0$ *be SUM-CG and let* c_u^{Γ} *and* $c_u^{\Gamma'}$ *the cost function of player* u *in* Γ *and* Γ'*, respectively. Then, for any strategy profile* S*, any player* u *and any* S_u*,* S'_u *strategies of* u *we have that* $c_u^{\Gamma'}(S_{-u}, S'_u) - c_u^{\Gamma'}(S_{-u}, S_u) = (c_u^{\Gamma}(S_{-u}, S'_u) - c_u^{\Gamma}(S_{-u}, S_u))/\lambda$

Proof. Let $\Gamma = \langle V, (w_u)_{u\in V}, \alpha, \beta\rangle$ be a Sum Celebrity Game, $S = (S_{-u}, S_u) \in \mathcal{S}(\Gamma)$, $S_u, S'_u \in \mathcal{S}(u)$ and $S' = (S_{-u}, S'_u)$.

Let $\Gamma' = \langle V, (w_u/\lambda)_{u\in V}, \alpha/\lambda, \beta\rangle$ with $\lambda > 0$. Let us show that the deviation of cost of the same player u and same strategy change is proportional to λ,

$$
\begin{aligned}
&c_u^{\Gamma'}(S') - c_u^{\Gamma'}(S) = \\
&= \frac{\alpha}{\lambda}|S'_u| + \sum_{\{v \mid d_{G[S']}(u,v)>\beta\}} \frac{w_v}{\lambda} - \left[\frac{\alpha}{\lambda}|S_u| + \sum_{\{v \mid d_{G[S]}(u,v)>\beta\}} \frac{w_v}{\lambda}\right] = \\
&= \frac{1}{\lambda}\left[\alpha|S'_u| + \sum_{\{v \mid d_{G[S']}(u,v)>\beta\}} w_v - \left[\alpha|S_u| + \sum_{\{v \mid d_{G[S]}(u,v)>\beta\}} w_v\right]\right] = \\
&= \frac{c_u^{\Gamma}(S') - c_u^{\Gamma}(S)}{\lambda}
\end{aligned}
$$

□

- Hence, by Proposition 9, regarding the behaviour and the evolution of any dynamics we can assume w.l.o.g that α is fixed so we must only address the multiple scenarios given by distinct weights of the players. Let $\alpha = 100$.

We separate the study between $\beta = 2$ and $\beta > 2$ and discuss similarities and differences between the case in which the weights are equal and in which they are different.

6.2.1 $\beta = 2$

The results obtained in the study of the relation between w and α together with the variation of the number of initial nodes can be found in the following figure.

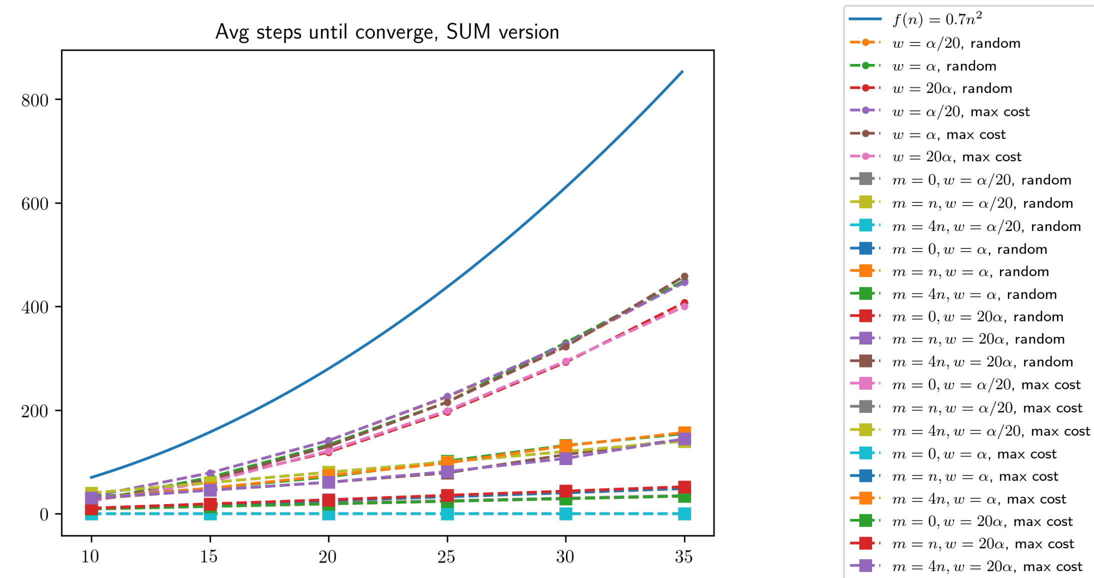


Figure 6.1: Experimental results for the Sum Greedy CG. Avg-steps, 50 trials

In the total of all the experiments with equal weights there are approximately 83% of deletions, 11% of swaps and 6% of additions. Usually the additions are made in early phases in which the diameter of the graph is reduced, once the nodes are close there is a phase of elimination of redundant links. Swaps are usually used in both phases. When the weights are different, the most used operator is still the delete but the swap becomes more important. In this case the proportion is 60% of deletions, 25% swaps and 15% additions. This is due to the fact that initially the links to lower weight nodes will be exchanged for links to higher weight nodes, taking into account the neighbors of each of them since $\beta = 2$.

It is observed how the main factor of the time of converge is the number of initial edges. When we do not fix the initial number of links, the expected number is $O(n^2)$, and we can see that quadratic tendency in the number of steps. On the other hand, when we establish the number of edges linear with n, this tendency is also linear. In addition, the necessary steps for $m = 4n$ are greater than for $m = n$ and the same behavior between $m = n$ and $m = 0$. Although the main factor is the one mentioned above, we can also observe how the max cost policy is the one that more frequently takes longer to converge comparing the same setting with the random policy. This tendency is maintained in experiments with different weights. Even though, in this case it seems to influence more the distribution of weights than the turn policy. This may be because heavier nodes guide the convergence process more clearly.

When we do not fix the number of initial edges or $m = 4n$ then the relation $w = \alpha$ is the slowest. This is due to the fact that a large number of swaps are effected even if they end up converging on the star graph. These swaps slow down the convergence process. When $m = n$ or $m = 0$ the slowest one is when $w = 20\alpha$. This is because it is cheaper to buy a link than to pay the weight of a node and large amount of additions are performed. When $m = 0$, the nodes are going to buy a link forming directly a star graph. In the case that $m = n$, we observe mainly a first phase of additions and swaps and then there is a second phase with mainly deletions reaching the star graph or at least a graph with diameter 2, as we expected because $\beta = 2$.

We have explored different weights for $n \leq 25$. We have observed in this case a fast convergence. Due to this different weight, at the beginning of the process the heavy nodes will have more links, in this way, the central nodes are better defined. The configurations that take longer are those that the weights are equidistant. This seems to be because the nodes swap links to higher weight nodes, but, as the higher weight nodes produce their changes then the previous nodes have to change again.

We have obtained along all the configurations and different n three diameters: 2, 3 or ∞, for equal and different weights. This matched the theoretical result of Nash equilibrium in which if the resulting graph was connected then $diam(G) \leq 2\beta + 1$. In addition, all graphs with infinity diameter have resulted in null graphs.

When the weights are identical, the infinite diameter is obtained in configurations where $m = 0$ and $w = \alpha$ or $w = \alpha/20$. Both cases were expected. Since when $w = \alpha$ the networks starts from the null graph and the cost of a link is the same that the weight of a node no one will have incentive to buy. In the second case, as a link is profitable only in the case it approach at least 20 nodes it was expected given the n we tested. When weights are different we obtain infinite when $m = 0$ and $w = T_3$, this distribution of weights is the only one that presents $w_{max} \leq \alpha$. This means that no node has an incentive to buy unless it gets closer to more than one node. As with $m = 0$ this cannot happen we get I_n.

As we know, the PoA of a SUM-CG when the NE graph is connected is $O(min\{\frac{n}{\beta}, \frac{W}{\alpha}\})$, so PoA is proportional to n.

In this case we get that r_{SUM-GE} of networks that are not I_n have been:

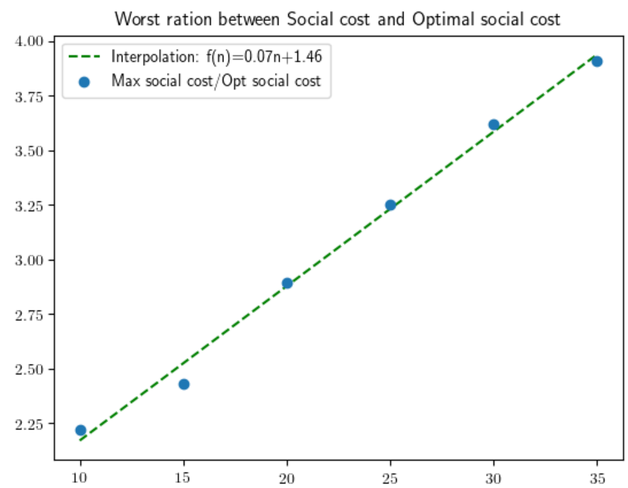


Figure 6.2: Experimental results for the Sum Greedy CG. Same and different weights. r_{SUM-GE} of all greedy equilibria that are not I_n

We observe again how the ratio is linear with respect to n, which confirms that the equilibrium moves away from the optimum as the number of nodes increase.

Networks in which the independent set is the resulting graph can be analyzed analytically. We do the study with equal weights:

- $w = \alpha$, $\frac{C(S)}{OPT(\Gamma)} = \frac{n(n-1)w}{min\{\alpha, n \cdot w\}(n-1)} = \frac{n(n-1)w}{min\{w, n \cdot w\}(n-1)} = \frac{n(n-1)w}{w(n-1)} = n$
- $w = \alpha/20$, $\frac{C(S)}{OPT(\Gamma)} = \frac{n(n-1)w}{min\{\alpha, n \cdot w\}(n-1)} = \frac{n(n-1)w}{min\{20w, n \cdot w\}(n-1)}$

  If $n \leq 20$, $\frac{n(n-1)w}{n \cdot w(n-1)} = 1$

  If $n > 20$, $\frac{n(n-1)w}{20w(n-1)} = \frac{n}{20}$

For the study of the topology we initially executed 50 times each configuration but we obtained that the *rl*, $w = 20\alpha$, *random* function had a peak at $n = 15$. The reason for this slow convergence was because the equilibrium was a star graph but in the intermediate process there were several nodes candidates to be the center of the star. This negatively affected convergence since there were more swaps and later deletes. To analyze if it was a configuration pathology or it was because of the random policy, we executed 1000 times each configuration, obtaining the following figure:

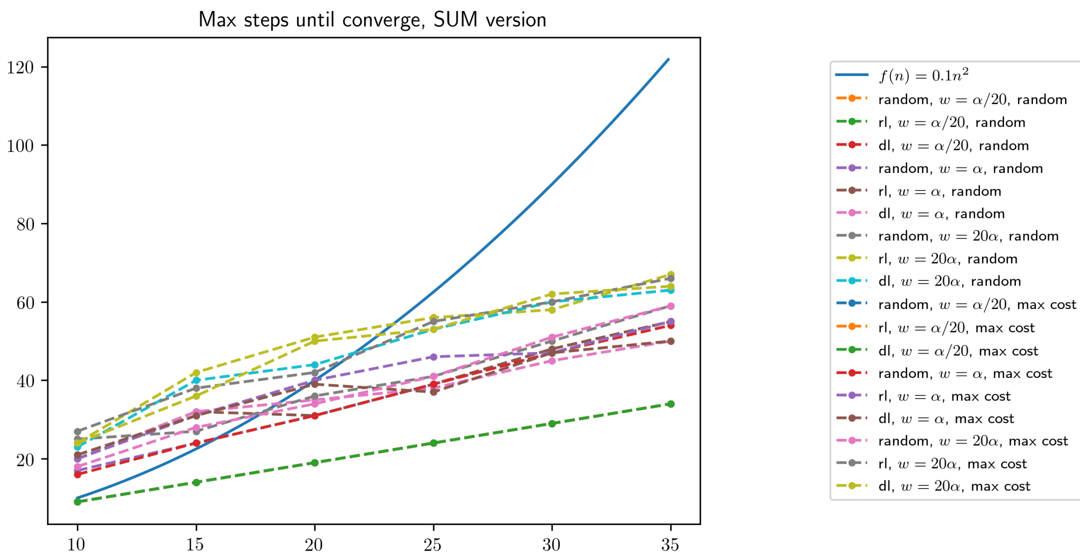


Figure 6.3: Experimental results for the Sum Greedy CG. Avg-steps, 1000 trials

It can be observed how all functions tend to grow linearly with the number of nodes and in this case we have not observed any strange peaks.

As we have seen before, the configuration of weights that took the longest to converge with $m = n$ was with $w = 20\alpha$. In this case, once again, this is the configuration that takes the longest.

In the case of different weights, we could expect that the center of the star, if produced, would be the node (or one of the nodes) of higher weight. Through our observations we have been able to verify that in most cases the heaviest node does not end up as the center of the star. Analyzing the traces we can observe how the main reason might be the turn policy, since nodes with lower weight due to the initial position in the network can accumulate more links. As for convergence time, similar values are obtained but with greater oscillations.

6.2.2 $\beta > 2$

So far we have made the study for $\beta = 2$. In this case we are going to reproduce the previous experiments but varying β in function of n. As we know $1 \leq \beta \leq n - 1$, for that reason we are going to use values of the form n/k, where $k > 1$. It has been taken into account that in the cost function β is equivalent to $\lfloor \beta \rfloor$ in order to correctly distribute the possible values.

The studied values of β are $n/2$, $n/3.5$ and $n/5$. The relation between w and α is the same as in previous experiments.

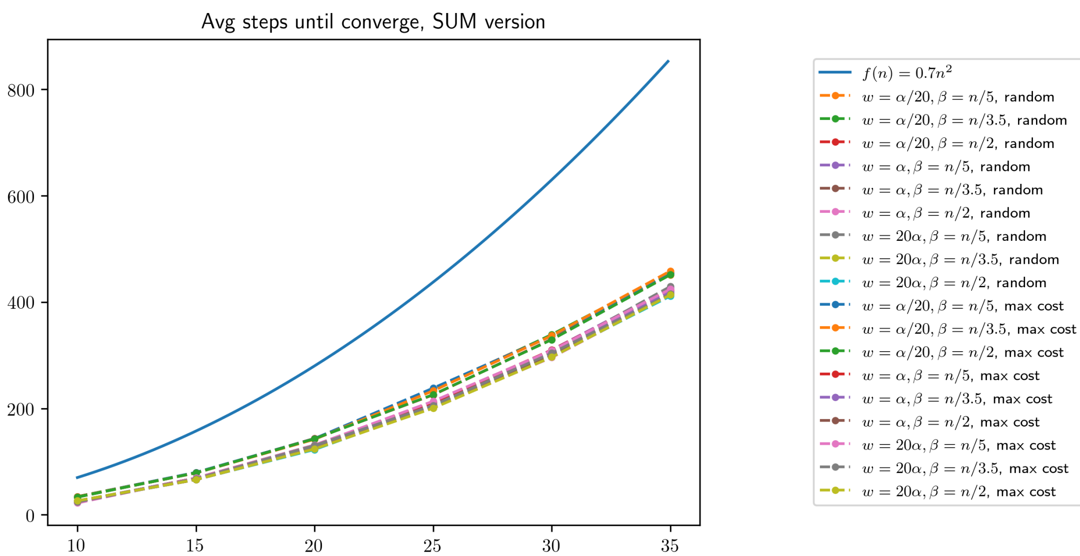


Figure 6.4: Experimental results for the Sum Greedy CG. Random initial graph. Avg-steps, 50 trials

Again, there is a tendency similar to the one obtained previously. The number of steps is mainly determined by the number of edges in the initial graph. In addition, the same behavior occurs when the configuration sets the number of initial edges.

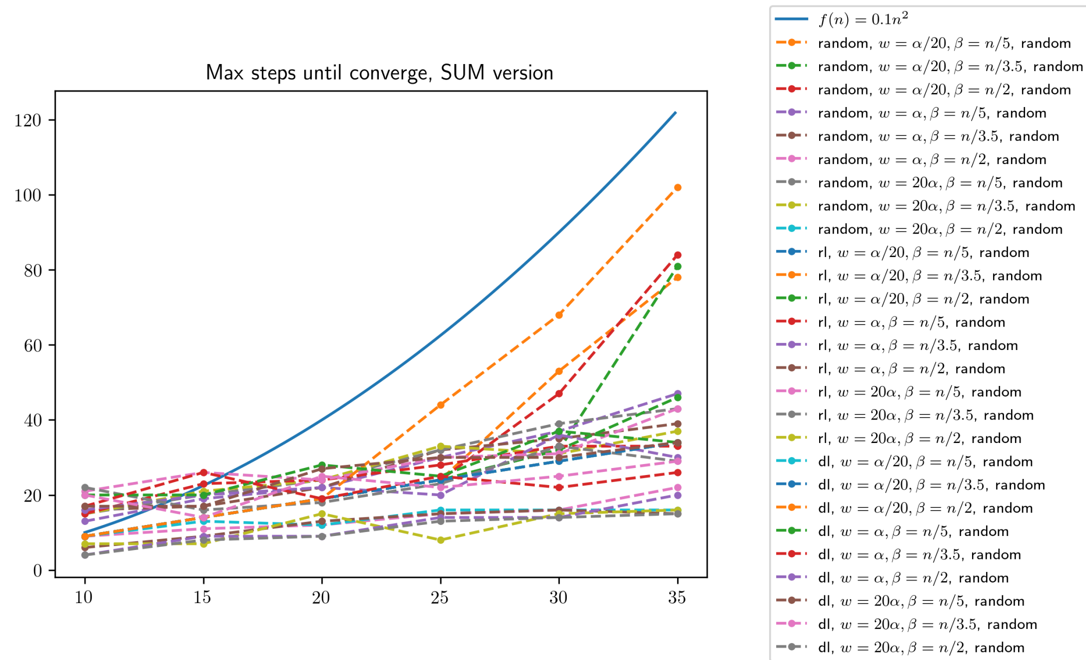
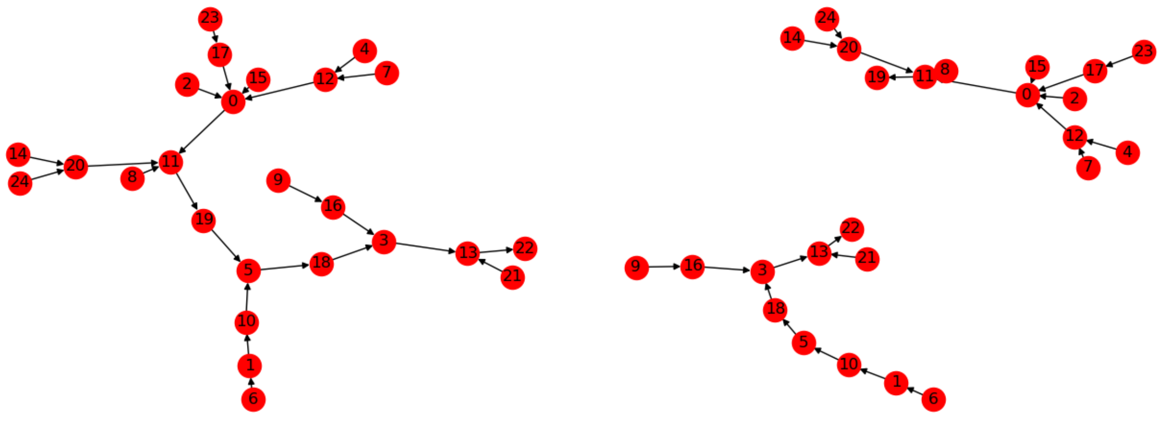


Figure 6.5: Experimental results for the Sum Greedy CG. The initial topology of the graph is *random*, *rl* or *dl*. Average number of steps needed to converge of 50 trials

For the case of the initial topology, β variation does affect the number of steps as can be seen in the figure below. In the case of different weights the behavior is similar. The configurations that tend to increase more than the rest are those with $w = \alpha/20$, that is, $w < \alpha$. In this case, a node will buy a link if, at a minimum, approach 20 nodes at a distance of at least β. This is more frequent as the number of nodes increases and as we increase the value of β since the number of nodes that must be approached to make it profitable remains constant. For this reason, more additions are produced and the number of steps required is higher.

We analyze the trace of *dl*, $w = \alpha/20$, $\beta = n/2$ with random turn policy to show the behavior. We use the case $n = 25$ to be able to display the nodes correctly.



$S^{(20)}$ $S^{(21)}$

Figure 6.6: Trace of: *dl*, $w = \alpha/20$, $\beta = n/2$ *random* with $n = 25$

We can see how in the transition from $S^{(20)}$ to $S^{(21)}$ the node 19 has removed the link and has formed two connected components. This node has removed the link because it only approached 11 nodes, so removing the link has improved since $w = \alpha/20$. For example, if the turn in state 20 had been for 5, a similar situation would have happened. On the other hand, in the early stages when a node generates several connected components it usually forms a component with few nodes that, in case it is the turn of one of them, it will swap in order to reconnect to the connected component with more nodes. As the process progresses, it is more likely that several components will be formed in which due to the cost of the link it will not be profitable to rejoin them. From this moment on, the network would start to carry out deletes until reaching the independent set.

These behavior also occurs for the max cost policy with $w = \alpha/20$. The reason is similar to the one mentioned above, but in this case all the equilibria end up being I_n. This is because the random allows more options of reconnection of the connected components that are formed.

In the case of different weights, the time of convergence show no apparent difference. Curiously, the case in which the convergence time is slower is due to the configuration of weights in which there is a more relevant node than the rest. In this configuration we get the following graph at equilibrium:

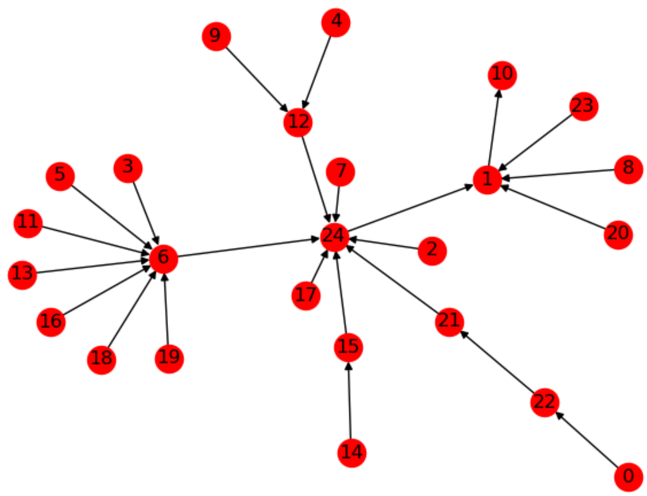


Figure 6.7: T_1, $\beta = 5$, $n = 25$

The reason for having a few additional steps is because the heaviest node (node 0) is in a leaf. In this way, small modifications of his connections forced to restructure multiple connections.

Among all the configurations tested with $\beta > 1$, we obtain that approximately 90% of the times the experiment with greater β converges faster. This is intuitively understood because with higher β the nodes need less number of links to approach a number of nodes and, as we have seen, the time of convergence is closely related to the number of links.

The diameters of the graphs still satisfy the property of the diameters of the Nash equilibrium.

In addition, r_{SUM-GE} decreases when n increases because β also increase. In the case of greedy equilibria that are not I_n we have:

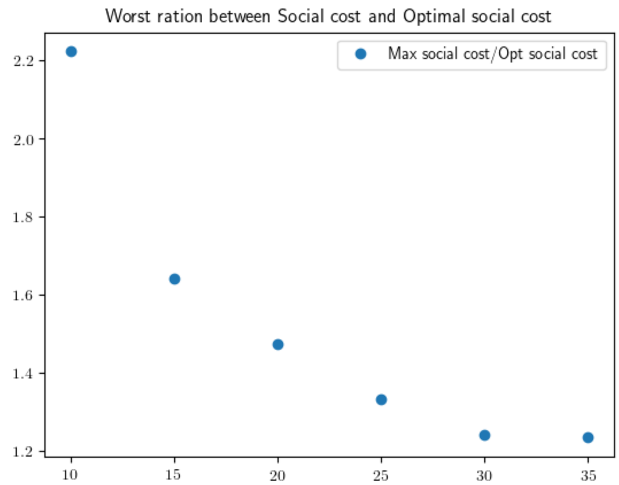


Figure 6.8: Experimental results for the Sum Greedy CG. r_{SUM-GE} of all greedy equilibria that are not I_n

Further remarks In all the experiments done for $\beta = 2$ the configurations have converged. Theoretically it is proven in the proposition 7 the existence of cycles in the Better Response SUM-CG dynamics and in [8] that the greedy model we analyze admits cycles. This indicates that, at least, these situations are uncommon.

We have observed how the determining factor of the steps needed is the number of initial edges, the greater the number of edges the greater the time needed.

It is observed how the greedy equilibria, despite being a super-set of the NE, preserves the properties of the maximum ratio between social cost and optimum social cost and the possible diameters of the equilibrium graphs.

In addition, when we have different weights we have observed a lower avg-steps but in the case of max-steps there are higher oscillations. This is due to the fact that the nodes that will concentrate the highest density of links are better defined. Even though, we can observe specific cases in which the initial network negatively affects the heavier nodes that should concentrate links and this causes a considerable slowdown.

6.3 Experimental study of Dynamics in Max Greedy Celebrity Games

In this section we analyze the same experiments we have done for the previous model but for the Max. In this case, we re-establish what our relation between w and α are according to the new cost function we are dealing with.

If we start from the independent set and the weights are the equal, the only way for a node to buy links will be that it buys all of them. Otherwise, the node would increase its cost because it continues to pay the weight of some node to which is not adjacent and also pay the cost of the links. In this case, the relation between α and w that keeps the cost identical is when $w = (n-1)\alpha$. Again, we are going to analyze a wider range of possibilities by multiplying by a factor greater and smaller than 1.

We consider the following scenarios:

- $n \leq 35$ when the weights are the same and $n \leq 25$, otherwise. The reason is computational limits.
-
  - All weight are identical w and are related to α as follows: $w = k(n-1)\alpha$ for $k = 1$, $k < 1$ and $k > 1$. We choose $k = 20$ to maintain the previously used constant.
  - The other possibility is that there are different types of weights: T_1, T_2, T_3 or T_4, where:
    * T_1: There is one node more important than the rest. We use as weight of the important node $w = 20(n-1)\alpha$ and for the rest $w = (n-1)\alpha$.
    * T_2: There are three types of nodes with approximately the same amount in each. The three possible weights are: $w = 20(n-1)\alpha$, $w = (n-1)\alpha$ and $w = (n-1)\alpha/20$.
    * T_3: The weights are distributed equidistant between $w_{min} = (n-1)\alpha/20$ and $w_{max} = (n-1)\alpha$.
    * T_4: The weights are distributed equidistant between $w_{min} = (n-1)\alpha/20$ and $w_{max} = 20(n-1)\alpha$.
- β parameter: $\beta = 2$ or $\beta > 2$ with: $\beta = n/2$, $\beta = n/3.5$ and $\beta = n/5$.
- Initial edge density: $m = 0$, $m = n$ and $m = 4n$.
- Initial topology: *random*, *rl* or *dl*.

As the weights we are going to study are in function of alpha we can prove the following property.

Proposition 10. *Let* $\Gamma = \langle V, (w_u)_{u \in V}, \alpha, \beta \rangle$ *and* $\Gamma' = \langle V, (w_u/\lambda)_{u \in V}, \alpha/\lambda, \beta \rangle$ *with* $\lambda > 0$ *be MAX-CG and let* c_u^{Γ} *and* $c_u^{\Gamma'}$ *the cost function of player* u *in* Γ *and* Γ'*, respectively. Then, for any strategy profile* S*, any player* u *and any* S_u*,* S'_u *strategies of* u *we have that* $c_u^{\Gamma'}(S_{-u}, S'_u) - c_u^{\Gamma'}(S_{-u}, S_u) = (c_u^{\Gamma}(S_{-u}, S'_u) - c_u^{\Gamma}(S_{-u}, S_u))/\lambda$

Proof. Let $\Gamma = \langle V, (w_u)_{u \in V}, \alpha, \beta \rangle$ be a Max Celebrity Game, $S = (S_{-u}, S_u) \in \mathcal{S}(\Gamma)$, $S_u, S'_u \in \mathcal{S}(u)$ and $S' = (S_{-u}, S'_u)$.

Let $\Gamma' = \langle V, (w_u/\lambda)_{u \in V}, \alpha/\lambda, \beta \rangle$ with $\lambda > 0$. Let us show that the deviation of cost of the same player u and same strategy change is proportional to λ,

$$
\begin{aligned}
&c_u^{\Gamma'}(S') - c_u^{\Gamma'}(S) = \\
&= \frac{\alpha}{\lambda}|S'_u| + \max_{\{v \mid d_{G[S']}(u,v) > \beta\}} \frac{w_v}{\lambda} - \left[\frac{\alpha}{\lambda}|S_u| + \max_{\{v \mid d_{G[S]}(u,v) > \beta\}} \frac{w_v}{\lambda}\right] = \\
&= \frac{1}{\lambda}\left[\alpha|S'_u| + \max_{\{v \mid d_{G[S']}(u,v) > \beta\}} w_v - \left[\alpha|S_u| + \max_{\{v \mid d_{G[S]}(u,v) > \beta\}} w_v\right]\right] = \\
&= \frac{c_u^{\Gamma}(S') - c_u^{\Gamma}(S)}{\lambda}
\end{aligned}
$$

□

- Hence, by Proposition 10, regarding the behaviour and the evolution of any dynamics we can assume w.l.o.g that α is fixed so we must only address the multiple scenarios given by distinct weights of the players. Let $\alpha = 100$.

6.3.1 $\beta = 2$

The results obtained in the study of the ratio of weight and alpha together with the variation of the number of initial nodes can be found in the following figure.

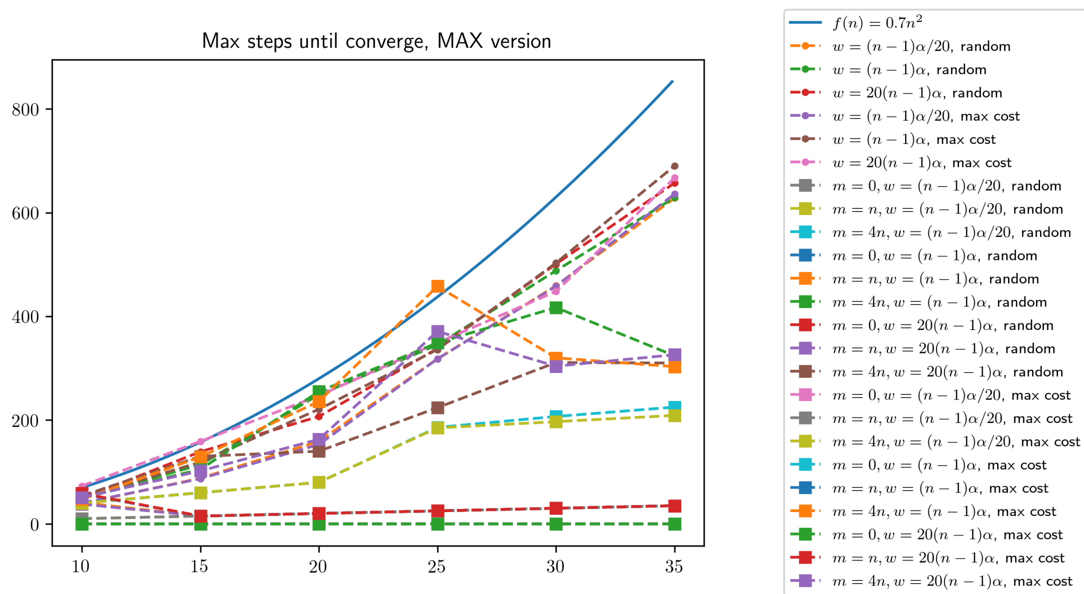


Figure 6.9: Experimental results for the Max Greedy CG. Max-steps, 50 trials

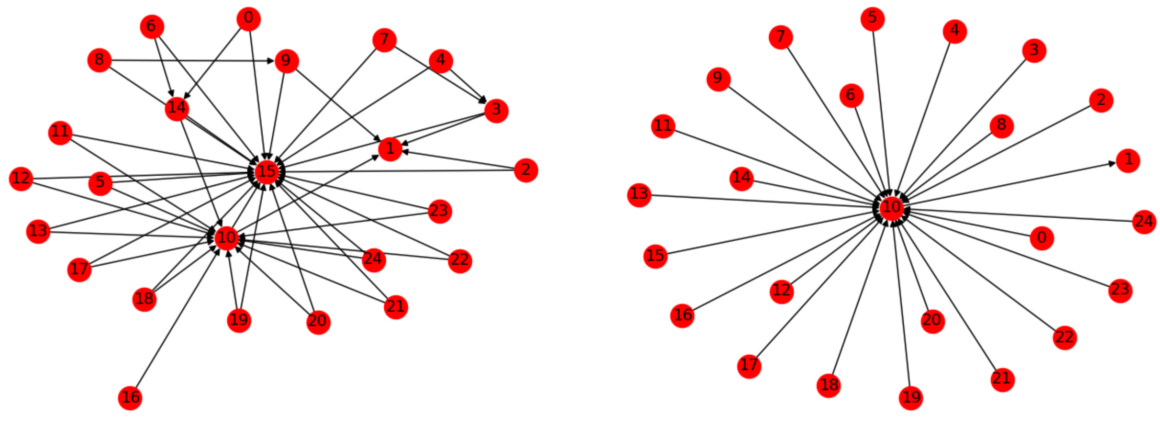
We have included the plot of max-steps since in the case of the average we get something similar to the Sum model: the configurations converge slower as more edges they initially have. Even though, in this model the different settings do not follow such a clear trend, aspect that can be observed clearly in $n = 25$. On the other hand, this distinction is clearly present in the average plot.

Again, the configurations with greater convergence time are those in which $\alpha < w$, those configurations are: $w = (n-1)\alpha$ and $w = 20(n-1)\alpha$. In this way nodes have incentive to buy links.

When we use different weights, the configurations that take longer to converge are those that there are different types of weight and in the initial phase is formed as the center of a possible star a node of weight significantly lower than the maximum weight. This causes that if a heavy node begins to concentrate link density there is a transition process that will involve a large number of steps. As n increases, it will be less likely that a low weight node concentrate a large number of links, making this behavior less common.

For the same weights or different weights, max cost turn requires more steps to converge. Furthermore, as the number of nodes decreases random and max cost are equaled.

We include a trace of the configuration with a peak in $n = 25$.



$S^{(400)}$ $S^{(458)}$

Figure 6.10: Trace of: $m = 4n$, $w = (n-1)\alpha$, *max cost* with $n = 25$

Analyzing the behavior in general, it is observed how the apparent reason of the peak of steps is due to the fact that there is not a single node as the center of the star and, due to the politics of the turn this problem is accentuated. Since there are several candidates for the center, transitions will be made between different nodes until the process reaches its equilibrium.

The proportion of additions, deletions and swaps used in this version has varied with respect to the previous version. In this case the deletions have been reduced to approximately 71% and the swaps increased to 25%, leaving the additions at 4%. Therefore, additions are less used than before. The increment of swaps might indicates that this version is more susceptible to slower convergence and possible configurations with more variable steps. When the weights are different the percentage loss of deletions is even more accentuated by considerably increasing the additions. The percentages are 55% deletes, 30% swaps and 15% additions. This is due to the fact that there are cases in which it is profitable to connect to the heaviest node to which you are not connected even paying the maximum of the remaining weights. This happens when there are nodes with very high weight compared to the rest.

When weights are equal in all configurations, we have obtained again three diameters exactly the same as in the previous version: 2, 3 and ∞. On the other hand, when we have different weights we also obtain graphs with diameter 4. Theoretically it is established that NE have $diam(G) \leq 2\beta + 2$, so again, we get that greedy equilibria fulfill this property of NE.

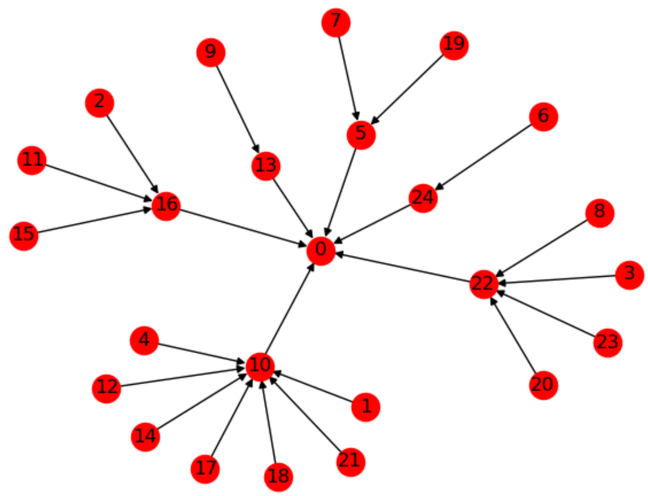


Figure 6.11: Example of $diam(G) = 4$. Configuration: $m = n$, T_1, random with $n = 25$

In this case the heaviest node is 0. We observe how the rest of the nodes are at a distance of at most 2 from the heaviest node and the diameter is 4.

In the study of the topology we have carried out the same procedure as in the previous version.

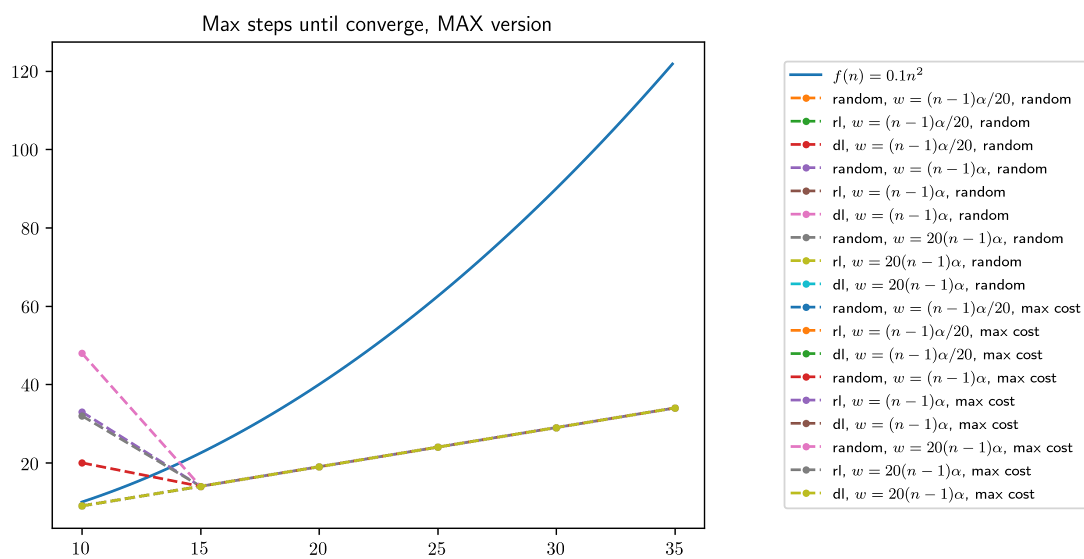
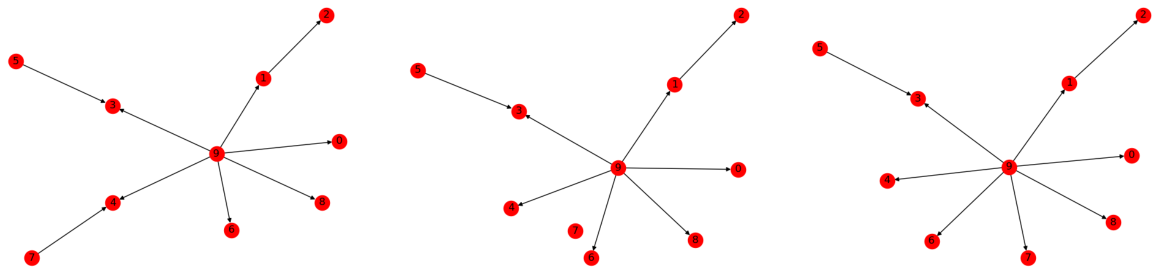


Figure 6.12: Experimental results for the Max Greedy CG. Avg-steps, 50 trials

In this occasion we observe a strange behavior, in which for $n = 10$ there are some configurations with a peak and that starting from $n = 15$ all converge directly to the independent set applying only deletion operators.



$S^{(4)}$ $S^{(5)}$ $S^{(6)}$

Figure 6.13: Trace of: *random*, $w = 20(n-1)\alpha$, *max cost* with $n = 10$

In this trace we can observe that node 9 is at a distance at most 2 of the rest, on the other hand, the rest of the nodes are at a greater distance due to node 2, 5 and 7. As a consequence the node 7 remove the link, since the cost of the link does not prevent it from paying w due to $\beta = 2$. In this case, the central node buys a link to node 7 since $\alpha < w$.

As we can see this situation is unlikely to occur as n increases, since the only way this happens is that buying a single link a node gets to have the rest at a distance at most β.

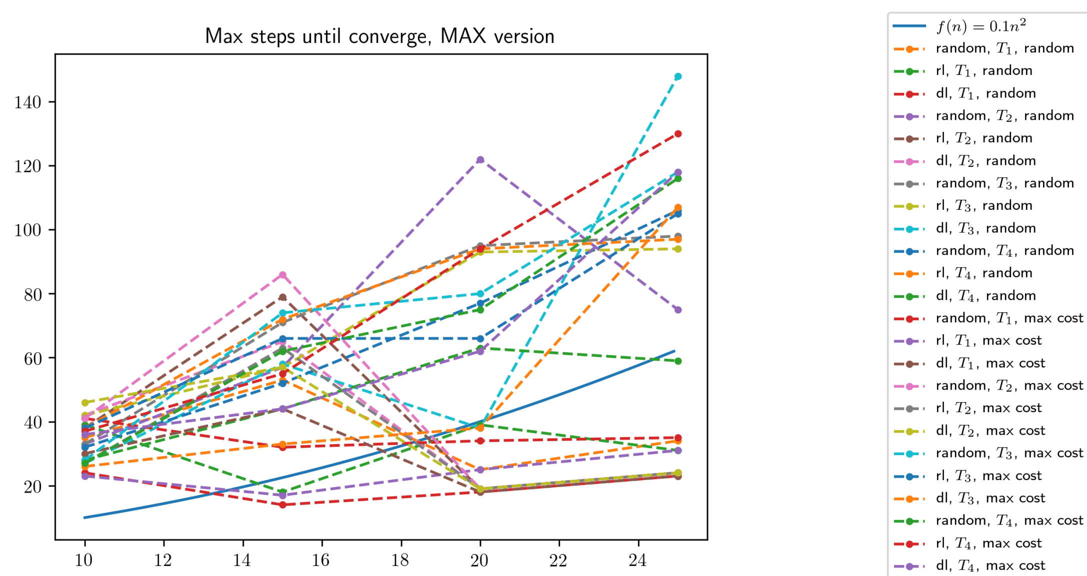


Figure 6.14: Experimental results for the Max Greedy CG. Avg-steps, 50 trials

On the other hand, with different types of weights the behavior is completely different. The configurations in which more steps are required are those in which there are nodes with weight much higher than α. In these cases the network will be connected and in addition different nodes can enter in dispute for the density of links either for their weight or for being better connected.

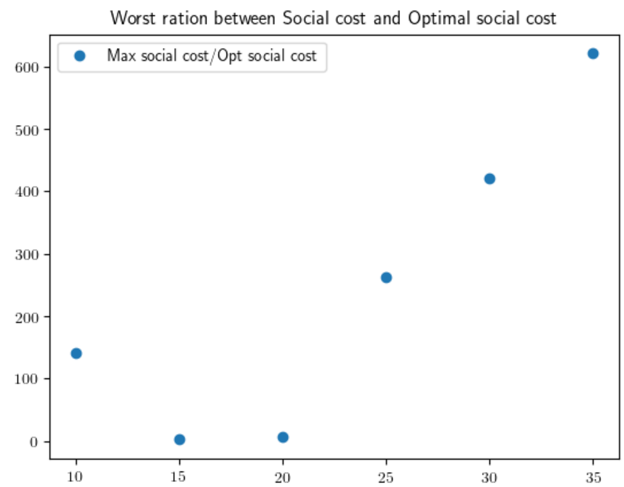


Figure 6.15: Experimental results for the Max Greedy CG, same weights. r_{MAX-GE} of all greedy equilibria that are not I_n

With different weights r_{MAX-GE} also increases as n increases because β also increases.

Theoretically we have that the price of anarchy of MAX-CG is $PoA \leq 2(w_{max}/\alpha)$ and $PoA(\Gamma) = O(n/\beta)$. As we can see, in networks where the independent set is not the final graph the PoA first presents a decrease and then an increase as the number of nodes increases. Even though, these two expressions are still satisfied for all configurations.

6.3.2 $\beta > 2$

For sake of presentation the plot only contain the results when the graph has no number of fixed initial edges.

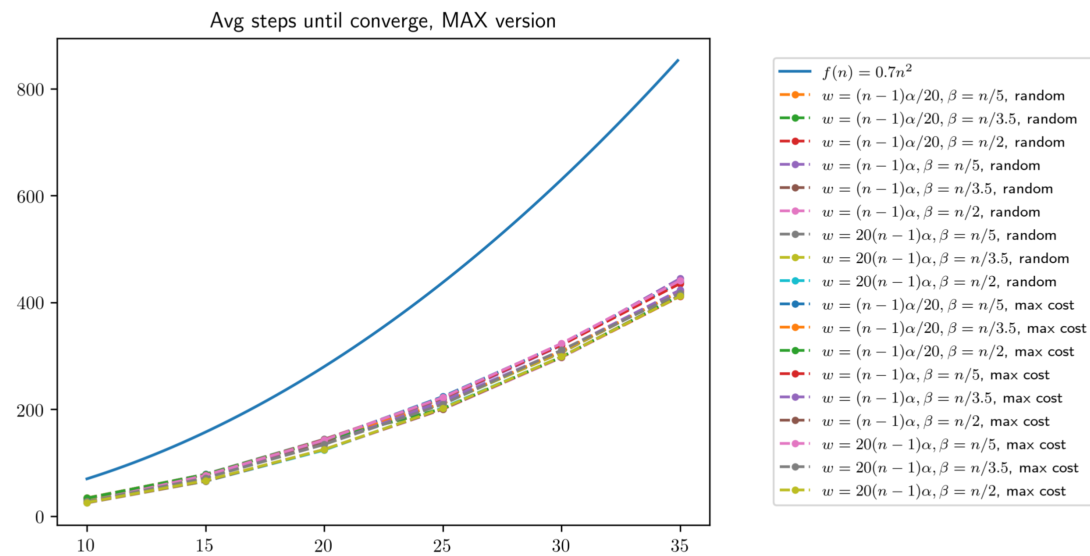


Figure 6.16: Experimental results for the Max Greedy CG. Random initial graph. Avg-steps, 50 trials

We are still getting the same results as before but this time it seems that when β is linearly related to n the avg-steps function is smoother. With different weights the behavior is the same as with equal weights.

We are going to analyze the initial topology. In the Sum version there were cases in which the number of steps increased considerably. On this occasion due to the nature of the maximum we hope that there are also these peaks.

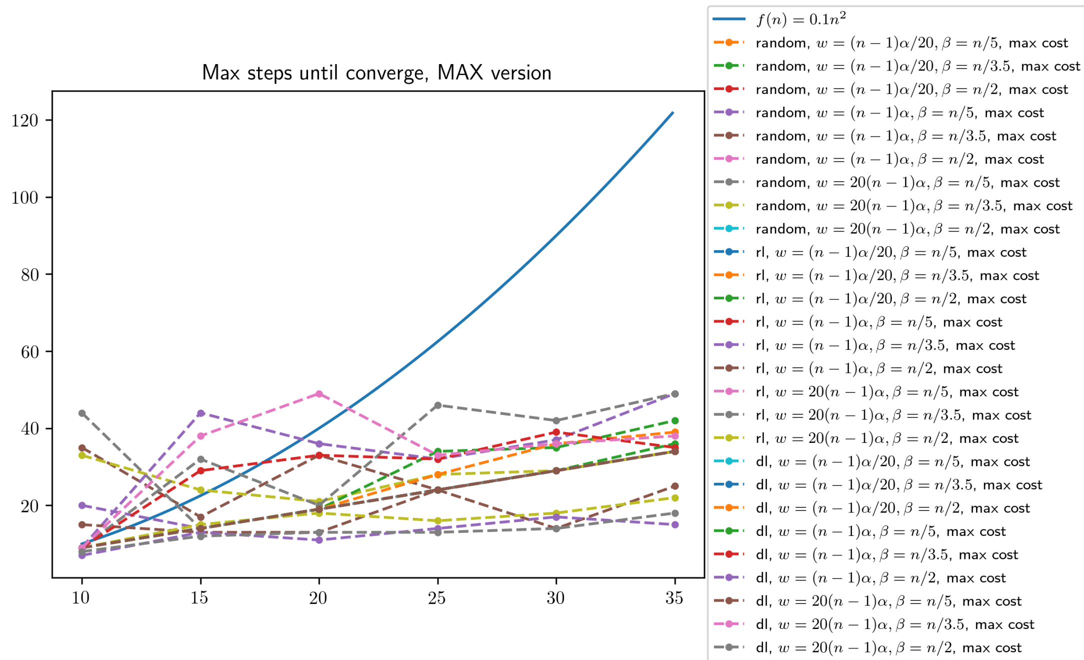


Figure 6.17: Experimental results for the Max Greedy CG. The initial topology of the graph is *random*, *rl* or *dl*. Average number of steps needed to converge of 50 trials

It is interesting to see that there is no configuration that stands out from the rest. Contrasting with this, we can observe that the necessary steps vary considerably. Even though, through the avg-steps can be observed how the trend is still linear with respect to n. With different weights the avg-steps behaves in a similar way, but in max-steps there are more oscillations. This is due to the same reason there were oscillations with $\beta = 2$.

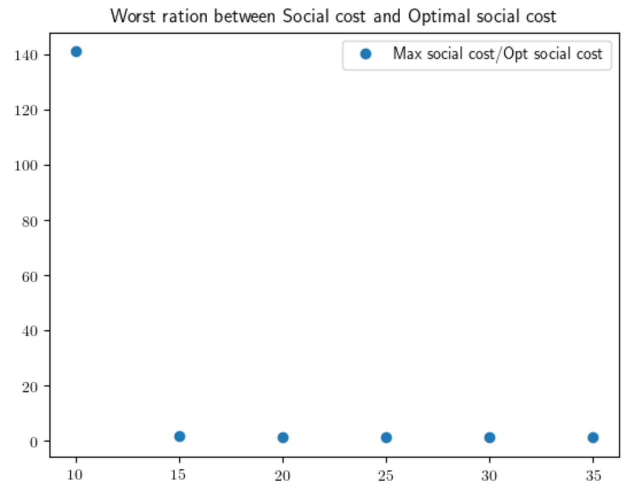


Figure 6.18: Experimental results for the Max Greedy CG. r_{MAX-GE} of all greedy equilibria that are not I_n

On this occasion r_{MAX-GE} it also decreases as we increase β. On this occasion the decrease is more abrupt reaching a minimum of approximately 1.15. When we use different weights there is also a tendency to lower the ratio.

Further remarks We have not found any cycle in any of the executions. In addition, the properties of the maximum ratio between social cost and optimum social cost as well as the diameter with respect to the theoretical properties of NE continue to be satisfied.

On this occasion, we have observed that different weights cause different behaviors compared to equal weights. This is due to the interaction between the heavy nodes and the nodes that accumulate more links. In addition, with different weights the network may take longer to converge if the node concentrating the highest density of links is not one of the heaviest nodes.

7 Conclusions and Future Work

The main objective of this project was to analyze the dynamics of Celebrity Games in order to understand better how decentralized and selfish networks without coordination among agents behave. Since agents act selfishly it is reasonable to think that they change their strategy to their best option. For that reason we have studied the Best Response dynamics.

Since the problem of computing a Best Response of a player is polynomial time solvable for $\beta = 1$ and NP-hard for $\beta > 1$, we have divided the research in two parts.

For $\beta = 1$, we have shown that for a Best Response Sum Celebrity Games dynamics it is not possible to obtain cycles and the problem of computing a Nash equilibrium is computable in polynomial time. On the other hand, for the Max model there is an initial configuration and turn policy that presents cycles but computing a Nash equilibrium is still polynomial time.

For β >1, since agents cannot afford exponential time for the response policy we have analyzed the Better Response. In this case the response is more flexible but it might be the case that we have to examine an exponential number of configuration. Furthermore, we have found that for any α and for any β there exist an initial configuration and a turn policy in which the Better Response dynamics have cycles in both models.

In order to perform an experimental study of a Better Response we have defined a greedy model, in which the possible new strategies are restricted. The new strategy of a player are those in which he adds, remove or swaps a link from his current strategy. In this model, the problem of computing a Best Response is polynomial time.

In the experiments of the Best Response Greedy Celebrity Games we have analyzed the number of steps rounds to converge. In both versions all tested configurations have converged in $O(n^2)$ where n is the number of nodes. This indicates that, despite the possibility of obtaining cycles, these pathological situations are very rare. In addition, we have verified that the structural properties obtained theoretically for Celebrity Games also verify for all the instances tested of the Greedy Celebrity Games. This indicates that, in an experimental way, the greedy equilibria is not worse in terms of the Price of Anarchy and that we have not obtained diameters greater than those expected in the Celebrity Games.

For future research, we consider that it would be interesting to analyze a random response policy, in which the player who has the turn generates a random strategy and makes the change of strategy if it improves its current cost. This response policy in the worst case will need an exponential number of rounds, but it would be interesting to study the expected time.

As a personal experience, I have been able to develop skills that I would not have been able to do otherwise. First of all, I had to learn the basic concepts of Game Theory and Network Creation Games. Furthermore, I had to learn how to read scientific articles and summarize the main ideas. In addition, I have trained skills gained during the degree such as complexity theory and programming. Overall, the project has been successful and my objective of taking part in a research project has been fulfilled.

Bibliography

[1] Alex Fabrikant, Ankur Luthra, Elitza Maneva, Christos H. Papadimitriou, and Scott Shenker. On a network creation game. In *Proceedings of the Twenty-second Annual Symposium on Principles of Distributed Computing*, PODC '03, pages 347–351, New York, NY, USA, 2003. ACM.

[2] Erik D. Demaine, Mohammadtaghi Hajiaghayi, Hamid Mahini, and Morteza Zadimoghaddam. The price of anarchy in network creation games. *ACM Trans. Algorithms*, 8(2):13:1–13:13, April 2012.

[3] C. Àlvarez, M. Blesa, A. Duch, A. Messegué, and M. Serna. Stars and celebrities: A network creation game. Technical report, ALBCOM - Algorismia, Bioinformàtica, Complexitat i Mètodes Formals, Jan 2015.

[4] C. Àlvarez, M. Blesa, A. Duch, A. Messegué, and M. Serna. Celebrity games. *Theoretical computer science*, 648:56–71, Oct 2016.

[5] C. Àlvarez and A. Messegué. Max celebrity games. In *Workshop on Algorithms and Models for the Web Graph*, pages 88–99, Dec 2016.

[6] Pascal Lenzner. *On selfish network creation.* PhD thesis, Humboldt University of Berlin, 2014.

[7] Bernd Kawald and Pascal Lenzner. On dynamics in selfish network creation. In Blelloch and Vöcking [11], pages 83–92.

[8] C. Àlvarez and A. Messegué. Dynamics in celebrity games. Personal Communication.

[9] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V. Vazirani. *Algorithmic Game Theory.* Cambridge University Press, New York, NY, USA, 2007.

[10] M. J. Osborne. *An introduction to game theory.* Oxford University Press, 2001.

[11] Guy E. Blelloch and Berthold Vöcking, editors. *25th ACM Symposium on Parallelism in Algorithms and Architectures, SPAA '13, Montreal, QC, Canada - July 23 - 25, 2013.* ACM, 2013.

A Project Management

A.1 Temporal planning

This section aims to describe the tasks that define the project, its estimated duration and the dependencies between tasks. In addition, it will explain how possible deviations will be solved and how it will affect the final duration of the project.

In all the tasks described above, except for the first one, the 3 roles of the project will participate: project director, PhD assistant and researcher.

We have divided the project into a series of tasks, so that we can identify on a large scale whether the objectives are being met.

A.1.1 Feasibility and study of concepts

Before deciding on this project, the feasibility and the basic concepts with which it was necessary to be familiar were analyzed with the project director. During the previous semester the material has been studied and several meetings have been held to track progress. The objective of this preliminary study was to focus the time of the project on the research itself.

As such, we will not count the time invested in this previous phase since it was carried out outside the project and it is difficult to determine.

A.1.2 Project planning

It can be divided into three sub-stages:

- Context and scope of the project.
- Temporal planning.
- Budget and sustainability.

The first two sub-stages are essential to carry out the project correctly, as otherwise we could be doing unnecessary work or in a disorganized way. The third sub-stage on the other hand, due to the characteristics of the project, can be combined with other tasks.

A.1.3 Main development

This is the main and most important task of the project. It covers the research, experimentation and writing the results of each of the parts.

In Celebrity Games the cost of a player is: the cost of creating the links plus and sum or max (depending on which model of the game is) of the weights of the agents that are farther away than the critical distance β.

We divide the project into the following parts:

Research of the Dynamics in Celebrity Games for $\beta = 1$

We will explore if the problem of computing a Nash equilibrium is polynomial time, in addition to studying whether it is possible to obtain cycles in the Best Response dynamics for both models.

Research of the Dynamics in Celebrity Games for $\beta > 1$

As computing a Best Response is NP-hard and agents cannot afford exponential time to change the strategy, they have to use an approximation of the best possible strategy change. For that reason, we are going to explore a greedy model in which the possible new strategies are restricted.

We are also going to explore if cycles are possible in a Better Response Celebrity Games.

Experimentation and result analysis of the Dynamics in Greedy Celebrity Games for $\beta > 1$

In this case we are going to analyze experimentally the greedy model introduced before. We are going to analyze the time of convergence, that is, the number of rounds needed until the network reach an equilibrium. In addition, we are going to compare the theoretical results of Celebrity Games with the greedy model, in order to examine how similar the equilibria are.

To test the convergence a set of graphs will be generated and its proper analysis will be done.

A.1.4 Final stage

At this stage we are going to write the documentation as well as preparing the final presentation.

Due to the nature of the project special attention is required. Therefore, considerable time will be devoted to writing, in this way the feedback given by the project manager can be taken into account.

A.2 Budget

To be able to carry out the project we will need resources. Below we will detail its cost according to the type of expense it belongs to.

For the calculation of amortization we have taken into account that the amortization time is 5 months.

A.2.1 Hardware resources

To develop the project we need a computer in which to program the code and perform the different tasks, either answer e-mails, consult articles, etc.

Product	**Price**	**Useful life**	**Amortization**
PC + peripherals	1200 €	4 years	125€
Total			125€

Table A.1: Hardware budget

A.2.2 Software resources

These are the main programs needed as indicated before. If we need any additional program we will try to use an open source alternative.

Product	**Price**	**Useful life**	**Amortization**
Windows 10	269 €	3 years	37,36€
LaTeX	0 €	-	0€
Python (+ libraries)	0 €	-	0€
git	0 €	-	0€
Total			37,36€

Table A.2: Software budget

A.2.3 Human resources

We break down the working hours for each task of the project and the role of those involved.

Role	Task				
	Project planning	$\beta = 1$	$\beta > 1$	Experimentation	Final stage
Project director	16 h	16 h	16 h	30 h	16 h
PhD Assistant	16 h	16 h	16 h	30 h	16 h
Researcher	90 h	90 h	90 h	130 h	50 h
Total	122 h	122 h	122 h	190 h	82 h

Table A.3: Estimated time per phase

Taking into account the different costs of each role we obtain the following human resources costs.

Role	Total work	€/hour	Total cost
Project director	94 h	40	3760 €
PhD Assistant	94 h	35	3290 €
Researcher	450 h	25	11250 €
Total	638 h		18300 €

Table A.4: Human resources budget

A.2.4 Unexpected costs

In case of any deviation in the planning we allocate a portion of the budget to possible overtime needed to complete the project. These hours belong to the margin time contemplated in the temporary planning.

Role	Hours	€/hour	Total cost
Project director	30 h	40	1200 €
PhD Assistant	30 h	35	1050 €
Researcher	30 h	25	750 €
Total	90 h		3000 €

Table A.5: Unexpected costs

A.2.5 Indirect cost budget

The average power we will need is approximately 200W. This includes the computer, lighting and items used from time to time such as the printer.

If we suppose that this is the average consumption, we can calculate that the total energy consumed will be the average power multiplied by the time we have invested in total, that is, the time obtained in human resources: 200W · 638h = 127,6 kWh.

Product	**Price**	**Units**	**Estimated cost**
Electricity	0,2 €/kWh	127,6 kWh	25,52 €
Internet	40 €	4 months	160 €
Office supplies	25€	1	25€
Total			210,52 €

Table A.6: Indirect costs

A.2.6 Total budget

Notice that a 10% of contingency has been added over the cumulative total in order to cover unexpected expenses that may occur during the course of the project.

The final breakdown of costs is presented below:

Concept	**Estimated cost**
Hardware resources	125 €
Software resources	37,36 €
Human resources	18300 €
Unexpected costs	3000 €
Indirect cost	210,52 €
Subtotal	21672,88 €
Contingency (10%)	2167,3 €
Total	23840,18 €

Table A.7: Total cost

A.3 Sustainability

A.3.1 Matrix ponderation

We will ponderate the matrix according to the standard seen in GEP.

The cells that belong to the development will have values between 0 and 10, instead the useful life will be from 0 to 20 and the risks from -20 to 0.

The final sum will be -60 to 90, with 90 being the highest possible sustainability score we can obtain.

	Development	**Exploitation**	**Risks**
Environmental	Design consumption	Ecological footprint	Environmental
	5	20	-5
Economic	Invoice	Viability plan	Economic
	5	20	-10
Social	Personal impact	Social impact	Social
	9	20	0
Total	19	60	-15
	64		

Table A.8: Sustainability Matrix

With a sustainability of 64 we can conclude that the project has an adequate sustainability index.

In the development phase the environmental and economic have been affected by the possibility of needing more resources than estimated, which would affect the ecological footprint and the budget. In the exploitation phase we get the maximum score because no extra resources are needed but the results are public at any time. The possible risks are the economic since we are not guaranteed to obtain an economic benefit from the project.

In the following, we will look more deeply at each of these parts.

A.3.2 Economical dimension

All costs related to the project have been detailed in the previous section. This description includes material costs, human resources and also takes into account possible unforeseen events.

The research generally has the same material needs as those described throughout the document and the only one that can vary is in the number of participants and their corresponding costs. In this case, the team carrying out the research is of a reduced size in which only one person participates for each role. In the case that the research was carried out by only the project director and his assistant, this would mean an increase in human costs, since the cost of each one is greater than the one of the researcher, and would also prevent them from carrying out other work.

A.3.3 Environmental dimension

The project does not have a major impact on the environment, as the main detrimental factor is indirect sources, such as the need for electricity and the necessary office supplies, such as sheets of paper for making notes.

The impact that electricity has on the environment depends on the supplier, as it will not have the same impact whether it comes from renewable energies or from some other source.

On the other hand the paper also depends on its origin, since it is not the same to use recycled paper that of another type.

In both cases, every effort will be made to minimize damage to the environment. In addition, indicate that this impact is common to all research projects.

A.3.4 Social dimension

This project will teach me what it means to be a researcher and help me decide my professional career.

Once the project is finished society in general will not notice any change, but the scientific community can use the results obtained for future research. Research similar to this project requires similar resources to those indicated above. This project will contribute to the study of a model from a point of view that had not been done before.

In research, apart from some projects with an immediate practical application, it is difficult to quantify their possible impact. In this case, the study is merely theoretical from which one can extract certain ideas regarding networking.

B Code for Simulations

B.1 Sum Celebrity Games

```
from random import choice, randint, sample

import networkx as nx
import numpy as np

class SCG:
    def __init__(self, w, a, B, edges=[]):
        self.w = w
        self.a = a
        self.B = B
        self.n = len(w)

        self.G = nx.Graph()
        self.G.add_nodes_from(range(self.n))
        self.G.add_edges_from((u, v, {'owner': u}) for u, v in edges)

    def generate_graph(self):
        """
        Generates a graph u.a.r
        No self loops nor multiedges
        """
        self.G.clear()
        self.G.add_nodes_from(range(self.n))
        for u in range(self.n):
            for v in range(u+1,self.n):
                if choice([True,False]):
                    x,y = sample([u,v],k=2);
                    self.G.add_edge(x, y, owner=x)

    def m_generate_graph(self, m):
        """
        Generates a graph u.a.r with 'm' edges
        No self loops nor multiedges
```

```
        """
        self.G.clear()
        self.G.add_nodes_from(range(self.n))
        for _ in range(m):
            while True:
                u = randint(0, self.n - 1)
                v = randint(0, self.n - 1)
                if u == v or self.G.has_edge(u, v):
                    continue
                self.G.add_edge(u, v, owner=u)
                break

    def random_line(self):
        self.G.clear()
        self.G.add_nodes_from(range(self.n))
        rp = np.random.permutation(self.n)
        for i in range(len(rp)-1):
            if choice([True,False]):
                self.G.add_edge(rp[i],rp[i+1],owner=rp[i])
            else:
                self.G.add_edge(rp[i+1],rp[i],owner=rp[i+1])


    def directed_line(self):
        self.G.clear()
        self.G.add_nodes_from(range(self.n))
        rp = np.random.permutation(self.n)
        for i in range(len(rp)-1):
            self.G.add_edge(rp[i],rp[i+1],owner=rp[i])


    def calc_cost(self, u):
        return self._calc_cost(u, sum)

    def _calc_cost(self, u, func):
        dist = nx.single_source_shortest_path_length(self.G, u, self.B)
        purchased_edges = 0
        weight_no_path = []
        for v in self.G.nodes:
            if u == v:
                continue
            if v not in dist: # if not in dist, distance(u,v) > B due
                cutoff
```

```
                weight_no_path.append(self.w[v])
            elif self.G.has_edge(u, v) and self.G[u][v]['owner'] == u: # v
                in S_u
                purchased_edges += 1
        return self.a * purchased_edges + func(weight_no_path)

    def is_NE(self):
        for u in range(self.n):
            br, _ = self.BestResponse(u)
            if not br:
                return False
        return True

    def socialcost(self):
        sc = 0
        for u in range(self.n):
            sc += self.calc_cost(u)
        return sc

    def BestResponse(self, u):
        """
        Check if strategy of 'u' is a BR and if not returns a u.a.r Best
           Response
        :return: (strategy 'u' is BR, u.a.r Better Response)
        """
        initial_cost = self.calc_cost(u)
        initial_edges = [(u, v, {'owner': u}) for v in self.G.neighbors(u)
           if
                        self.G[u][v]['owner'] == u]
        self.G.remove_edges_from(initial_edges)

        not_buy_nodes = [v for v in self.G.neighbors(u) if
                        self.G[u][v]['owner'] == v] + [u]
        poss_nodes = [v for v in range(self.n) if v not in not_buy_nodes]

        min_cost = initial_cost
        BRs = []
        for ss in SCG._powerset(poss_nodes):
            act_edges = [(u, v, {'owner': u}) for v in ss]
            self.G.add_edges_from(act_edges)
            cost = self.calc_cost(u)
            self.G.remove_edges_from(act_edges)
            if min_cost > cost:
```

```
            min_cost = cost
            BRs = [ss]
        elif cost == min_cost:
            BRs.append(ss)

    self.G.add_edges_from(initial_edges)
    if min_cost == initial_cost:
        return True, None
    return False, choice(BRs)

"""
GRAPH FUNCTIONS
"""

def add_edges(self, u, nodes):
    for v in nodes:
        self.G.add_edge(u, v, owner=u)

def del_edges(self, u, nodes):
    for v in nodes:
        self.G.remove_edge(u, v)

def nod_noadj(self, u):
    """
    Nodes not adjacent to 'u'
    """
    return [v for v in range(self.n) if v not in self.G.neighbors(u)
       and u != v]

def out_edges(self, u):
    """
    Nodes to which 'u' has purchased a link. S_u
    """
    res = []
    for v in range(self.n):
        if self.G.has_edge(u, v) and self.G[u][v]['owner'] == u:
            res.append(v)
    return res

"""
UTILS
"""
```

```
    @staticmethod
    def _powerset(s):
        x = len(s)
        for i in range(1 << x):
            yield [s[j] for j in range(x) if (i & (1 << j))]
```

B.2 Max Celebrity Games

```
from .SCG import SCG

class MCG(SCG):
    def calc_cost(self, u):
        return self._calc_cost(u, lambda x: max(x, default=0))
```

B.3 Greedy Celebrity Games

```
from random import choice, shuffle
from enum import Enum

from model.SCG import SCG
from model.MCG import MCG

class Policy(Enum):
    RANDOM = 0
    MAXCST = 1

class Version(Enum):
    SUM = 0
    MAX = 1

class GreedyCG():
    def __init__(self, w, a, B, edges=[], policy=Policy.RANDOM, ver=
       Version.SUM):
        if Version.SUM == ver:
            self.Game = SCG(w, a, B, edges)
        else:
            self.Game = MCG(w, a, B, edges)
        self.policy = policy
```

```
def generate_graph(self):
    self.Game.generate_graph()

def m_generate_graph(self, m):
    self.Game.m_generate_graph(m)

def random_line(self):
    self.Game.random_line()

def directed_line(self):
    self.Game.directed_line()


def check_best_add(self, u, current_cost):
    return self._check_best_move(u, self.Game.nod_noadj,
                                 self.Game.add_edges,
                                 self.Game.del_edges, current_cost)

def check_best_del(self, u, current_cost):
    return self._check_best_move(u, self.Game.out_edges,
                                 self.Game.del_edges,
                                 self.Game.add_edges, current_cost)

def _check_best_move(self, u, get_nodes, func, func_inv, current_cost)
    :
    nodes_exam = get_nodes(u)
    min_cost = current_cost
    nodes = []
    for v in nodes_exam:
        func(u, [v])
        c = self.Game.calc_cost(u)
        func_inv(u, [v])
        if c < min_cost:
            min_cost = c
            nodes = [v]
        elif c == min_cost:
            nodes.append(v)
    if min_cost == current_cost:
        return False, current_cost, None
    return True, min_cost, choice(nodes)

def check_best_swp(self, u, current_cost):
```

```
        O = self.Game.out_edges(u) # u owns edge to elements of O
        N = self.Game.nod_noadj(u) # u not own edge to elements of N
        min_cost = current_cost
        change = []
        for v in O:
            for w in N:
                self.Game.del_edges(u, [v])
                self.Game.add_edges(u, [w])
                c = self.Game.calc_cost(u)
                self.Game.del_edges(u, [w])
                self.Game.add_edges(u, [v])
                if c < min_cost:
                    min_cost = c
                    change = [(v, w)]
                elif c == min_cost:
                    change.append((v, w))
        if min_cost == current_cost:
            return False, current_cost, None
        return True, min_cost, choice(change)

    def perform_local_search_step(self, u, change_strategy, current_cost):
        """
        True -> 'u' can improve the cost
        """
        dec_add, c_add, move_add = self.check_best_add(u, current_cost)
        dec_del, c_del, move_del = self.check_best_del(u, current_cost)
        dec_swp, c_swp, move_swp = self.check_best_swp(u, current_cost)

        if dec_add or dec_del or dec_swp:
            move = None
            if not change_strategy:
                return True
            c_min = min(c_add, c_del, c_swp)
            if c_min == c_del:
                self.Game.del_edges(u, [move_del])
                move = ('del',u,move_del)
            elif c_min == c_swp:
                self.Game.del_edges(u, [move_swp[0]])
                self.Game.add_edges(u, [move_swp[1]])
                move = ('swp',u,(move_swp[0],move_swp[1]))
            else:
                self.Game.add_edges(u, [move_add])
                move = ('add',u,move_add)
```

```
            return True, move
        return False, None

    def turnPolicy(self):
        cost_list = []
        for u in range(self.Game.n):
            cost = self.Game.calc_cost(u)
            cost_list.append((u,cost))
        if self.policy == Policy.RANDOM: # random order
            shuffle(cost_list)
            return cost_list
        else: # max cost order
            return sorted(cost_list, key=lambda x : x[1])


    def executeDynamics(self):
        steps = 0
        Stable = False
        trace = []
        while not Stable:
            Stable = True
            node_list = self.turnPolicy() # list of (node,cost)
            while node_list:
                u, c = node_list.pop() # pop last element of list
                changed, move = self.perform_local_search_step(u, True, c)
                if changed:
                    trace.append(move)
                    steps += 1
                    Stable = False
                    if self.Game.n**2 < steps:
                        with open('log.txt','a') as f:
                            f.write('Did not converge in n^2 steps\n')
                    break
        return steps, trace
```